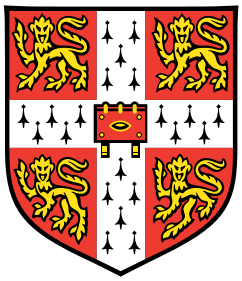
Continuous Dimensional Emotion Tracking in Music

Vaiva Imbrasaitė

April 2015



University of Cambridge
Computer Laboratory

Downing College

This dissertation is submitted for the degree of Doctor of Philosophy.

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text.

It is not substantially the same as any that I have submitted, or, is being concurrently submitted for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University of similar institution except as declared in the Preface and specified in the text

This dissertation does not exceed the regulation length of 60 000 words, including tables and footnotes.

Summary

The size of easily-accessible libraries of digital music recordings is growing every day, and people need new and more intuitive ways of managing them, searching through them and discovering new music. Musical emotion is a method of classification that people use without thinking and it therefore could be used for enriching music libraries to make them more user-friendly, evaluating new pieces or even for discovering meaningful features for automatic composition.

The field of Emotion in Music is not new: there has been a lot of work done in musicology, psychology, and other fields. However, automatic emotion prediction in music is still at its infancy and often lacks that transfer of knowledge from the other fields surrounding it. This dissertation explores automatic continuous dimensional emotion prediction in music and shows how various findings from other areas of Emotion and Music and Affective Computing can be translated and used for this task.

There are four main contributions.

Firstly, I describe a study that I conducted which focused on evaluation metrics used to present the results of continuous emotion prediction. So far, the field lacks consensus on which metrics to use, making the comparison of different approaches near impossible. In this study, I investigated people's intuitively preferred evaluation metric, and, on the basis of the results, suggested some guidelines for the analysis of the results of continuous emotion recognition algorithms. I discovered that root-mean-squared error (RMSE) is significantly preferable to the other metrics explored for the one dimensional case, and it has similar preference ratings to correlation coefficient in the two dimensional case.

Secondly, I investigated how various findings from the field of Emotion in Music can be used when building feature vectors for machine learning solutions to the problem. I suggest some novel feature vector representation techniques, testing them on several datasets and several machine learning models, showing the advantage they can bring. Some of the suggested feature representations can reduce RMSE by up to 19% when compared to the standard feature representation, and up to 10-fold improvement for non-squared correlation coefficient.

Thirdly, I describe Continuous Conditional Random Fields and Continuous Conditional Neural Fields (CCNF) and introduce their use for the problem of continuous dimensional emotion recognition in music, comparing them with Support Vector Regression. These two models incorporate some of the temporal information that the standard bag-of-frames approaches lack, and are therefore capable of improving the results. CCNF can reduce RMSE by up to 20% when compared to Support Vector Regression, and can increase squared correlation for the valence axis by up to 40%.

Finally, I describe a novel multi-modal approach to continuous dimensional music emotion recognition. The field so far has focused solely on acoustic analysis of

songs, while in this dissertation I show how the separation of vocals and music and the analysis of lyrics can be used to improve the performance of such systems. The separation of music and vocals can improve the results by up to 10% with a stronger impact on arousal, when compared to a system that uses only acoustic analysis of the whole signal, and the addition of the analysis of lyrics can provide a similar improvement to the results of the valence model.

Acknowledgements

First of all, I would like to thank the Computer Laboratory for creating a wonderful place to do your PhD in. The staff have always been more than accommodating with all of our requests and extremely helpful in any way they could be. I would like to thank the Student Admin and Megan Sammons in particular for being passionate about undergraduate teaching and creating opportunities for us to contribute. I want to thank the System Administrators and Graham Titmus in particular for being ever so helpful with any sort of technical problems. And also my biggest thanks to Mateja Jamnik for making women@CL happen and all the opportunities and support that it provides.

I am grateful for my research group—I feel very lucky to have chosen it, and to have been surrounded by a group of amazing people. Our daily coffee breaks were an anchor in my life and I will miss them greatly. I would like to give special thanks to Ian, for keeping me sane through the write-up, Tadas, as without him some of the work described here could never have happened, Leszek, for so many things that would not fit on this page, Luke, for many interesting conversations, Marwa, for being a wonderful officemate and for comparing notes on writing up and to everyone else in the group—you are all great. Many thanks to Alan for interesting discussions, Neil, for always making my day a little brighter, and, of course, my supervisor Peter Robinson for all the opportunities he created for me.

My viva turned out to be a lot less scary and a lot more enjoyable than I expected. I would like to thank my examiners, Neil Dodgson and Roddy Cowie, for a very interesting discussion about my work. The feedback I got helped me put my work in perspective and make some important improvements to this dissertation. Your comments were greatly appreciated!

I would also like to thank my first IT teacher Agnė Zlatkauskienė, who spotted me in our extracurricular classes and decided to teach a 12-year old girl to program. If not for her, I might not be where I am now—the world would be a better place if there were more teachers like her.

Finally, I would like to thank my parents, who failed to tell me that Mathematics and then Computer Science are not girly subjects, and who supported me all the way through. I hope I have made you proud.

Contents

Figures

Tables

Introduction

1

1.1. Motivation and approach

Music surrounds us every day, and with the dawn of digital music that is even more the case. The way the general population approaches music has greatly changed, with the majority of people buying their music online or using online music streaming services [BPI, 2013]. The vastly bigger and easily accessible music libraries require new, more efficient ways of organising them, as well as better ways of searching for old songs and discovering new songs. The increasingly popular field of Affective Computing [Picard, 1997] offers a solution—tagging songs with their musical emotion. Bainbridge et al. [2003] have shown that it is one of the natural descriptors people use when searching for music, thus providing a user-friendly way of interacting with music libraries. Musical emotion can also be used to evaluate new pieces, or to discover meaningful features that could be used for automatic music composition among other things.

The focus of my work is automatic continuous dimensional emotion tracking in music. The problem lends itself naturally to a machine learning solution and in this dissertation I show a holistic view of the different aspects of the problem and its solution. The goal of my PhD is not a perfect system with the best possible performance, but a study to see if and how findings in other fields concerning emotion and music can be translated into an algorithm, and how the individual parts of the solution can affect the results.

Automatic emotion recognition in music is a very new field, which opens some very exciting opportunities, as not a lot of approaches have been tried. However, it also lacks certain guidelines, making the work difficult to compare across different researchers. The contents of this dissertation reflect both of these observations—there are chapters that focus on showing the importance of certain methodological techniques and setting some guidelines, and other chapters that focus on trying out new approaches and showing how they can affect the results. The experimental conditions were kept as uniform as possible throughout the dissertation by the use of the same dataset, feature-set and distribution of songs for cross-validation, allowing a good and direct comparison of the results.

1.2. Research areas and contributions

There are four main focus areas covered in this dissertation: the use of different evaluation metrics to measure the performance of continuous musical emotion prediction systems, the techniques that can be employed to build feature vectors, different machine learning models that can be used, and the different modalities that can be exploited to improve the results.

Evaluation is a very important part of developing an algorithm, as it is essential to compare the performance of a system with either its previous versions or other systems in the field. Unfortunately, as continuous dimensional emotion recognition is a new field, there are no agreed-upon guidelines as to which evaluation metrics to use, making comparisons across the field difficult to make. A significant part of my work was focused on analysing the differences between different evaluation metrics and identifying the most appropriate techniques for evaluation (Chapter 3). To achieve that, I designed and executed a novel study to find out which of the most common evaluation metrics people intuitively prefer and I was able to suggest certain guidelines based on the results of the study.

The second focus area of my work was translating certain findings from the field of Emotion in Music into techniques for feature representation, as part of a machine learning solution to continuous dimensional emotion recognition in music. A lot of work gets put into either the design of features themselves or into the machine learning algorithms, but the feature vector building stage is often forgotten. In Chapter 4, I suggest several new feature representation techniques, all of which were able to achieve better results than a simple standard feature representation, and some of which were able to improve the results by up to 18.6% as measured by root-mean-squared error (RMSE), and several times as measured by correlation. The main ideas behind the different suggested representation techniques include: expectancy–the idea that an important source of musical emotion is in violation of or conformance to listener's musical expectations–and the fact that different musical features take different amounts of time to affect listener's perception [Schubert, 2004].

Another important factor in perception of emotion in music is its temporal characteristics. A lot of continuous emotion prediction techniques take the bag-of-frames approach, where each sample is taken in isolation and its relationship with previous and future samples is ignored. I addressed this problem in two distinct ways: through feature representation techniques (Chapter 4) and by introducing two machine learning models which incorporate some of that lost temporal information (Chapter 5). Both approaches gave positive results. Including surrounding samples into the feature vector for the current sample greatly reduced root mean squared error, and resulted in a large increase (several times) in the average song correlation. The two new machine learning models were also highly beneficial—Continuous Conditional Neural Fields in particular was often the best-performing model when compared to Continuous Conditional Random Fields and Support Vector Regression (SVR), improving the results by reducing RMSE by up to 20% and increasing average squared correlation for valence models by up to 40%.

The final part of my work is concerned with building a multi-modal solution to the problem of continuous dimensional emotion prediction in music (Chapter 6). While some of the static emotion prediction systems exploit some aspects of multi-modality of the data, the majority of emotion prediction systems (and continuous solutions in particular) employ acoustic analysis only (see Section 2.2 for examples). I built a multi-modal solution by splitting the input into three: I suggested separating the vocals from the background music, and analysing the two signals separately, as well as analysing lyrics and including those features into the solution. To achieve this, I used several music and voice separation techniques, and had to develop an analysis of lyrics that would be suitable to a continuous emotion prediction solution, as most of the other sentiment analysis in text systems focus on larger bodies of text. Combining the three modalities into a single system achieved the best results witnessed in this dissertation: when compared to the SVR acoustic only model, average song RMSE is decreased by up to 23% for arousal and by up to 10% for valence; CCNF is affected less, and the results are improved by up to 6%.

1.3. Structure

This dissertation is divided into 7 chapters, described below.

- **Chapter 2: Background** Emotion recognition in music is an intrinsically interdisciplinary problem, and at least a basic understanding of the relevant psychology, musicology and the general area of affective computing is essential to produce meaningful work. Chapter 2 begins with an introduction to Affective Computing and its main subfields, describing in more detail problems that are similar to emotion recognition in music and the techniques that my approach borrows. It then delves deeper in the topic of music emotion recognition, giving an in-depth analysis of the problem and all the components required to define and solve it.
- **Chapter 3: Evaluation metrics** As the field of dimensional emotion recognition and tracking is still fairly new, the evaluation strategies for the various solutions are not well defined. Chapter 3 describes the issues that arise when evaluating such systems, explores the different evaluation techniques and describes a study that I undertook to determine the most appropriate evaluation metric to use. It also covers the guidelines I suggest based on these results.
- **Chapter 4: Feature vector engineering** Chapter 4 introduces the machine learning approach to emotion recognition in music—describing the features that I have used for the models, and various feature vector building techniques that I have developed and tested.
- **Chapter 5: Machine learning approaches** The investigation of the different machine learning models that I used for this problem are described in Chapter 5. Here I justify the use of the Radial Basis kernel for Support Vector Regression and show the need of correct selection of hyper-parameters used in training. I also describe two new machine learning models—Continuous Conditional Random Fields and Continuous Conditional Neural Fields—that have never

been used for music emotion recognition before. I compare the models using several datasets and several feature representation techniques.

– **Chapter 6: Multi-modality** Chapter 6 shows a new, multi-modal approach to emotion recognition in music. The first section describes music-voice separation and its use for music emotion recognition. The second section is focused on sentiment analysis from lyrics. Finally, the third section combines the two together into a single system.

– **Chapter 7: Conclusions** This dissertation is concluded with Chapter 7 which summarises the contributions and their weaknesses and identifies the future work areas relevant to this problem.

1.4. Publications

Vaiva Imbrasaitė, Peter Robinson Music emotion tracking with Continuous Conditional Neural Fields and Relative Representation. *The MediaEval 2014 task: Emotion in music. Barcelona, Spain, October 2014*. Chapter 5

Vaiva Imbrasaitė, Tadas Baltrušaitis, Peter Robinson CCNF for continuous emotion tracking in music: comparison with CCRF and relative feature representation. *MAC workshop, IEEE International Conference on Multimedia, Chengdu, China, July 2014*. Chapter 5

Vaiva Imbrasaitė, Tadas Baltrušaitis, Peter Robinson What really matters? A study into people's instinctive evaluation metrics for continuous emotion prediction in music. *Affective Computing and Intelligent Interaction, Geneva, Switzerland, September 2013*. Chapter 3

Vaiva Imbrasaitė, Tadas Baltrušaitis, Peter Robinson Emotion tracking in music using continuous conditional random fields and baseline feature representation. *AAM workshop, IEEE International Conference on Multimedia, San Jose, CA, July 2013*. Chapters 4 and 5

Vaiva Imbrasaitė, Peter Robinson Absolute or relative? A new approach to building feature vectors for emotion tracking in music. *International Conference on Music & Emotion, Jyväskylä, Finland, June 2013*. Chapter 4

James King, Vaiva Imbrasaitė Generating music playlists with hierarchical clustering and Q-learning. *European Conference on Information Retrieval, Vienna, Austria, April 2015*.

Background 2

In the last twenty years, there has been a growing interest in automatic information extraction from music that would allow us to manage our growing digital music libraries more efficiently. With the birth of the field of Affective Computing [Picard, 1997], researchers from various disciplines became interested in emotion recognition from multiple sources: facial expression and body posture (video), voice (audio) and words (text). While work on emotion and music had been done for years before Affective Computing was defined [Scherer, 1991; Krumhansl, 1997; Diaz and Silveira, 2014], it certainly fueled multidisciplinary interest in the topic [Juslin and Sloboda, 2001, 2010]. The first paper on automatic emotion detection in music by Li and Ogihara [2003] was published just over 10 years ago, and since then the field has been growing quite rapidly, although there is still a lot to be explored and a lot of guidelines for future work to be set. Music emotion recognition (MER) should be seen as part of the bigger field of Affective Computing, therefore should learn from and share with other subfields. It must also be seen as part of the more interdisciplinary field of Music and Emotion, as only by integrating with other disciplines can its true potential be reached.

2.1. Affective Computing

Affective Computing is a relatively new field. It is growing and diversifying quickly, and now covers a wide range of areas. It is concerned with a variety of objects that can "express" emotion—starting with emotion recognition from human behavior (facial expressions, tone of voice, body movements, etc.), but also looking at emotion in text, music and other forms of art.

Affective computing also requires strong interdisciplinary ties: psychology for emotion representation, theory and human studies; computer vision, musicology, physiology for the analysis of emotional expression; and finally machine learning and other areas of computer science for the actual link between the source and the affect.

There is also a wide range of uses for techniques developed in the field—starting from simply improving our interaction with technology, to enriching music and text libraries, helping people learn, and preventing accidents by recognizing stress, aggression, etc.

2.1.1. Affect representation

There are three main emotion representation models that are used, to varying degree, in different fields of affective computing (and psychology in general)—categorical, dimensional and appraisal.

The categorical model suggests that people experience emotions as discrete categories that are distinct from each other. At its core is the Ekman and Friesen [1978] categorical emotion theory which introduces the notion of basic or universal emotions that are closely related to prototypical facial expressions and specific physiological signatures (increase in heart rate, increased production in sweat glands, pupil dilation, etc.). The basic set of emotions is generally accepted to include joy, fear, disgust, surprise, anger and sadness, although there is some disagreement between psychologists about which emotions should be part of the basic set. Moreover, this small collection of emotions can seem limiting, and so it is often expanded and enriched with a list of various complex emotions (e.g. passionate, curious, melancholic, etc.). At the other extreme, there are taxonomies of emotion that aim to cover all possible emotion concepts, e.g. the Mindreading taxonomy developed by Baron-Cohen et al. [2004] that contains 412 unique emotions, including all the emotion terms in English language, excluding only the purely bodily states (e.g. hungry) and mental states with no emotional dimension (e.g. reasoning). Consequently, the accepted set of complex emotions varies greatly both between different subfields of affective computing and also between different researcher groups (and even within them)—a set of 280 words or standard phrases tend to be used quite frequently, but the combined set of such words spans around 3000 words of phrases [Cowie et al., 2011]. Moreover, the categorical approach suffers from a potential problem of different interpretations of the labels by people who use them (both the researchers and the participants) and this problem is especially serious when a large list of complex emotions is used.

Dimensional models disregard the notion of basic (or complex) emotions. Instead, they attempt to describe emotions in terms of affective dimensions. The theory does not limit the number of dimensions that is used—it normally ranges between one (e.g. arousal) and three (valence, activation and power or dominance), but four and higher dimensional systems have also been proposed. The most commonly used model was developed by Russell [1980] and is called the circumplex model of emotion. It consists of a circular structure featuring the pleasure (valence) and arousal axes—Figure 2.1 shows an example of such a model with valence displayed on the horizontal and arousal on the vertical axis. Each emotion in this model is therefore a point or an area in the emotion space. It has the advantage over the categoric approach that the relationship and the distance between the different emotions is expressed in a much more explicit manner, as well as providing more freedom and flexibility. The biggest weakness of this model is that some emotions that are close to each other in the arousal-valence (AV) space (e.g. angry and afraid, in the top left corner) in real life are fundamentally different [Sloboda and Juslin, 2010]. Despite this, the two dimensions have been shown by Eerola and Vuoskoski [2010] to explain most of the variance between different emotion labels and more and more people adopt this emotion representation model, especially

because it lends itself well to computational models.

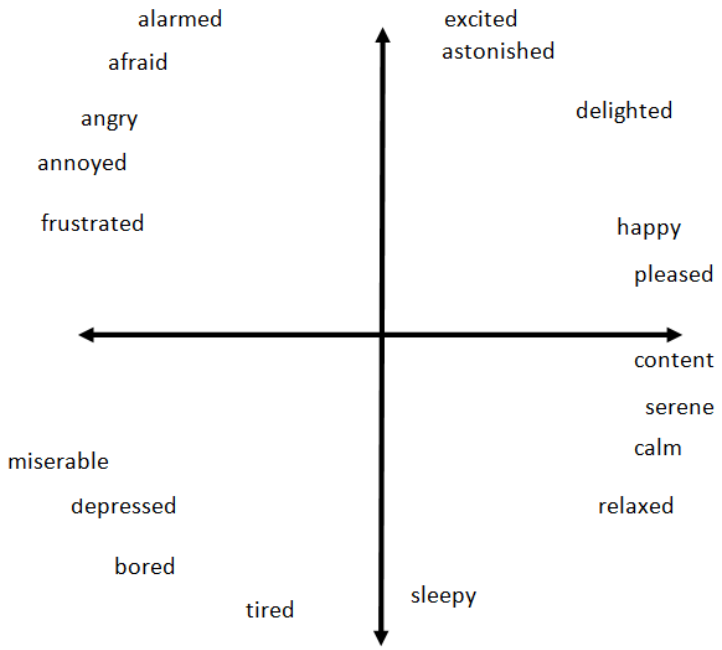


Figure 2.1: Image of the arousal-valence space with the appropriate emotion labels, based on Russell [1980].

The most elaborate model of all is the appraisal model proposed by Scherer [2009]. It aims to mimic cognitive evaluation of personal experience in order to be able to distinguish qualitatively between different emotions that are induced in a person as a response to those experiences. In theory, it should be able to account for different personalities, cultural differences, etc. At the core of this theory, there is the component process appraisal model, which consists of five components: cognitive, peripheral efference, motivational, motor expression and subjective feeling. The cognitive component appraises an event, which will potentially have a motivational effect. Together with the appraisal results, this will affect the peripheral efference, motor expression and subjective feeling. These, in turn, will have an effect on each other and also back to the cognitive and motivational components. Barthet and Fazekas [2012] showed that the model allows both for the emotions that exhibit a physiological outcome, and those that are only registered mentally. Despite the huge potential that this model has, it is more useful as a way of explaining emotion production and expression and it less suitable for computational emotion recognition systems due to its complexity.

2.1.2. Facial expression analysis

One the best researched problems within the field of affective computing is emotion recognition from facial expressions. The main idea driving this field is that there is a particular facial expression (or a set of them) that gets triggered when an emotion is experienced, and so detecting these expressions would result in a detection of the emotion. Similarly to the field of music emotion recognition (see Section

2.2), most of the work is done on emotion classification using categorical emotion representation approaches, mostly based on the set of basic emotions (which is where the association of emotion with a distinctive facial expression comes from), but there is a recent trend to move towards dimensional emotion representation.

As the task of emotion (or facial expression) recognition is rather difficult, the initial attempts to solve it included quite important restrictions on the input data: a lot of the time the person in the video would have to be facing the camera (fixed pose) and the lighting had to be controlled as well. In addition to that, as natural expressions are reasonably rare and short-lived, the majority of databases used for facial expression recognition contain videos of acted emotions (see Zeng et al. [2009] for an overview). Given the growing body of research which shows that posed expressions exhibit different dynamics and use different muscles from the naturally occurring ones, and with the improvements in the state of the art, there is an increasing focus on naturalistic facial expression recognition.

There are two main machine learning approaches to facial expression recognition: one based on the detection of affect itself using a variety of computer vision techniques, and another based on facial action unit (AU) recognition as an intermediate stage. The Facial Coding System has been developed by Ekman and Friesen [1978], and it is designed to objectively describe facial expressions in terms of distinct AUs or facial muscle movements. These AUs were originally linked with the set of the 6 basic emotions, but can also be used to describe more complex emotions. The affect recognition systems tend to be based on either geometric features (shape of facial components, such as eyes, mouth, etc., or their location, usually by tracking a set of facial points) or on appearance features (such as wrinkles, bulges, etc., approximated using Gabor wavelets, Haar features, etc.), with the best systems using both. Recently, there appeared a few approaches trying to incorporate 3D information (e.g. Baltrušaitis et al. [2012]) which allow a less pose-dependent system to be built.

With the increase in the complexity of data used for this problem, increasingly complex machine learning models had to be employed to be able to exploit the information available. Notable examples of these are Nicolaou et al. [2011] who used Long Short-Term Memory Recurrent Neural Networks that enable the recurrent neural network to learn the temporal relationship between different samples, Ramirez et al. [2011] who used Latent-Dynamic Conditional Random Fields to model the multi-modality of the problem (incorporating audio and visual cues) as well as the temporal structure of the problem. Similar to these, Continuous Conditional Random Fields [Baltrušaitis et al., 2013] and Continuous Conditional Neural Fields [Baltrušaitis et al., 2014] have been developed in our research group to be capable of dealing with the same signals and are especially suited for continuous dimensional emotion prediction.

2.1.3. Emotion in voice

Another important area of affective computing is speech emotion recognition, which can also be seen as the field most related to music emotion recognition. While the early research on both music emotion and the research on emotion re-

cognition from facial expressions tended to focus on the set of basic emotions, a lot of work on emotion recognition from speech is done on recognition of stress. One of the main applications for this problem is call centres and automatic booking management systems that could detect stress and frustration in a customer's voice and adapt their responses accordingly. The use of dimensional emotion representation is becoming more popular in this field as well, although as in music emotion recognition (MER, Section 2.2.1), researchers have reported better results for the arousal axis than for the valence axis [Wöllmer et al., 2008].

The applications for emotion recognition from speech also allow for relatively easy collection of data (while the labelling is as difficult as for the other affective computing problems)—the samples can be collected from recorded phone conversations at call centres and doctor-patient interviews as well as radio shows. Unfortunately that also raises a lot of privacy issues, so while the datasets are easy to collect and possibly label, it is not always possible to share them publicly, which makes the comparison of different methods more difficult. This leads to a lot of the datasets being collected by inviting people to a lab and asking them to act emotions out. To mitigate the problem of increased intensity of acted emotions, non-professional actors are often used (see Ververidis and Kotropoulos [2006] for a survey of datasets), though it has been argued that while acted emotions are more intense, that does not change the correlation between various emotions and the acoustic features, just their magnitude. However, similarly to other Affective Computing fields, the focus of speech emotion recognition systems is shifting towards non-acted datasets [Eyben et al., 2010], and we are even starting to see challenges based on such corpora [Schuller et al., 2013].

El Ayadi et al. [2011] groups the features used in emotion recognition in speech into 4 groups: continuous, qualitative, spectral and TEO- (Teager energy operator) based features. The arousal state of the speaker can affect the overall energy and its distribution over all frequency ranges and so it can be correlated with continuous (or acoustic) speech features. Such features can be grouped into 5 groups: pitch-related features, formants features, energy-related features and timing and articulation features, and can often also include various statistical measures of the descriptors. Qualitative vocal features, while related to the emotional content of speech, are more problematic to use, as their labels are more prone to different interpretations between different researchers and are generally more difficult to automatically extract. The spectral speech features, on the other hand, tend to be easy to extract and constitute a short-time representation of the speech signal. The most common of these are linear predictor coefficients, one-sided autocorrelation linear predictor coefficients, mel-frequency cepstral coefficients (MFCC)—some of these have been adopted and heavily used by music emotion recognition researchers. Finally, the nonlinear TEO-based features are designed to mimic the effect of muscle tension on the energy of airflow used to produce speech—the feature focuses on the effect of stress on voice and has successfully been used for stress recognition in speech, but has not been as successful for general emotion recognition in speech.

2.1.4. Sentiment analysis in text

Text is another common channel for emotion expression and therefore a good input for affect recognition. Opinion mining (or sentiment analysis) takes up a large part of this field—analyzing reviews to measure people's attitudes towards a product and mining blog posts and comments in order to predict who is going to win an election are just two attractive examples of potential applications. The line between affect recognition in text and sentiment analysis is thin, since most of the time the sentiment polarity (how positive/negative the view is) is what interests us most (see Pang and Lee [2008] for a survey on Sentiment Analysis in text).

Even though at first sight it might seem that text analysis should be easier than emotion recognition from, for example, facial expressions, there are a lot of subtleties present in text that make this problem hard to solve. A review might express a strong negative opinion even though there are no ostensibly negative words occurring in it (e.g. "If you are reading this because it is your darling fragrance, please wear it at home exclusively, and tape the windows shut."). Another problem is the separation between opinions and facts (which is much more of an issue in sentiment analysis than in affect recognition)—"the fact that" does not guarantee that an objective truth will follow the pattern and "no sentiment" does not mean that there is not going to be any sentiment either.

The dependence of sentiment extraction on context and domain is even more problematic, since most of the time the change in context and/or domain is not followed by a change in vocabulary (and therefore in the set of keywords). Finally, it also suffers from the general natural language processing issues, such as irony, metaphor, humour, complex sentence structures, and, in the case of web and social media texts, grammar and spelling mistakes and constantly changing vocabulary (e.g. words like "sk8er", "luv", etc.).

As in general affect recognition, both categorical and dimensional approaches in sentiment (or emotion) representation are used, as well as various summarization techniques for opinion mining. Dimensional representation for sentiment analysis is usually limited to just the valence axis (see Pang and Lee [2008] for a list of approaches using it), but for standard affect recognition two or three axes can be used, for example in the work by Dzogang et al. [2010].

Techniques

The techniques employed in the models for sentiment analysis can generally be grouped into several categories.

One of the most common approaches is the Vector Space Model (VSM, a more detailed description of it is provided in Section 6.2.1). Different term-weighting techniques exist (such as Boolean weighting, Frequency weighting, Term Frequency-Inverse Document Frequency weighting, etc.), and Pang et al. [2002] showed that for sentiment analysis, Boolean weighting might be more suitable than Frequency weighting which is generally more used in Information Retrieval models.

While the VSM takes a complete bag-of-words approach to the problem, more context-aware techniques can help improve the accuracy of such systems. A common technique used in the field is n-grams (see Section 6.2.1 for more details), where phrases are used instead of individual words. Bigrams are used most frequently and Pang et al. [2002] has shown that compared to unigrams, systems employing bigrams can achieve better performance.

Another technique borrowed from the general Natural Language Processing field is part-of-speech (POS) tagging. One of the main reasons why POS tags are used for sentiment analysis systems is that they provide a basic word sense disambiguation. Combining unigrams with their POS tags can improve the performance of a sentiment analysis system, as shown by Pang et al. [2002].

A more syntax-aware system enables some understanding of valence shifters: negations, intensifiers and diminishers. Kennedy and Inkpen [2006] have shown that the use of valence shifters can increase both the accuracy of an unsupervised model (when an affective dictionary is used) as well as a supervised machine learning model by creating bigrams (a pair of a valence shifter and unigram). More commonly, just the negation is used, either by inverting the valence value of a term based on an affective dictionary or by attaching "NOT" to a term to create another term with the opposite meaning.

Another popular approach is to map a simple VSM based on simple terms or n-grams into a topic model, where each axis represents a single topic rather than a single term. Topic models are usually generated through statistical (unsupervised) analysis of a large collection of documents. It is essentially a dimensionality reduction technique based on the idea that each word can be described by a combination of various implicit topics. It fixes the problem of synonymy that a simple VSM suffers from—now two synonyms would be positioned close to each other in the topic space as opposed to being perpendicular to each other in a VSM. Mullen and Collier [2004] have shown that incorporating such data into the feature vector improves the performance of a machine learning model used to determine the positivity of movie or music reviews.

Dataset of affective norms

An important tool that is often used for sentiment analysis in text is an affective norms dictionary. A well designed, publicly available and reliable dictionary of words that are labeled with their valence, arousal and, sometimes, dominance values can not only help compare different approaches to emotion or sentiment recognition in text, but can also be used for other purposes: research into emotions themselves, the effect of emotional features on word processing and recollection, as well as automatic emotion estimation for new words.

One of the most widely used dictionaries of affective norms is the Affective Norms for English Words (ANEW) collection. Developed by Bradley and Lang [1999], ANEW is a dictionary containing 1034 English words with three sets of ratings for each: valence, arousal and power. The emotional ratings were given by a group of Psychology class students and collected in a lab, in groups ranging between 8 and 25 students, balanced for gender. The dataset contains not only the aggregated

mean ratings for each dimension, as well as their standard deviation values, but also the same set of values separated by gender. While the dataset contains only just over 1000 words, it has been widely used in both sentiment analysis in prose, as well as emotion recognition in music via the analysis of lyrics.

An updated version of ANEW has recently been published by Warriner et al. [2013]. In their dataset, Warriner et al. [2013] have valence, arousal and dominance ratings for 13915 English lemmas. Unlike ANEW, this dictionary was compiled through an online survey of American residents through the Amazon Mechanical Turk (MTurk)[1] service. The dataset contains most of the words present in ANEW, and has been shown to have good correlation with the mean ratings both in ANEW and several other dictionaries both in English and other languages. Interestingly, the valence ratings had substantially higher correlation (of around 0.9 or higher) than those for arousal or dominance (between 0.6 and 0.8, if present). Similarly to other datasets (including ANEW), the Warriner dataset shows a V-shaped correlation between arousal and valence, and arousal and dominance, and a linear correlation between valence and dominance, suggesting that the dominance axis might not be of much use to sentiment analysis. It also shows a positive emotion bias in the valence (and dominance) ratings for the words appearing in the dictionary, which corresponds to the Pollyanna hypothesis proposed by Boucher and Osgood [1969], that suggests that there is a higher prevalence of words associated with a positive emotion rather than a negative one. As the Warriner dictionary contains a substantially larger set of words, and can easily be extended through a simple form of inflectional morphology, it has a lot more potential to be useful for both sentiment analysis in text in general, and work described in this dissertation.

There are also some dictionaries and some tools that are specialised for a particular subset of sentiment analysis in text. One example of such a system is SentiStrength[2]. SentiStrength was developed by Thelwall et al. [2010] as a tool for sentiment recognition in short, informal text, such as MySpace comments. They used 3 female coders to give ratings for over 1000 comments, with simultaneous ratings on both the positive and the negative scale—i.e. instead of using a single valence axis, it was split into two. While focusing only on the valence axis is a common approach in this field, Thelwall et al. [2010] noticed that people who participated in the labeling study treated expressions of energy, or arousal, as amplifiers for the positivity or the negativity of a word, and that expressions of energy were considered as positive, unless the context indicated that they were negative. Thelwall et al. [2010] started off with a manually coded set of words' ratings which were then automatically fine-tuned to increase the classification accuracy. SentiStrength is therefore a context-specific tool for automatic sentiment recognition in text, but it provides researchers with a fast and convenient tool that can be used in a growing field that focuses on short, informal text, such as comments or Twitter messages.

[1] http://mturk.com

[2] http://sentistrength.wlv.ac.uk/

2.2. Music emotion recognition

Emotion recognition in music might seem like a small, well defined field at first glance, but it actually is a multi-faceted problem that needs careful definition if one is to have any hope of being able to compare different approaches. Figure 2.2 and the following sections describe some of the main choices that one needs to make in order to define the problem he or she is about to tackle. While some of the choices (e.g. first level choice of the source of features) will only affect the processing (and the data) required for the actual system (which can also be argued to change its type), the choice of the level of granularity or the representation of the emotion will change the approach completely. I will refer to and emphasise these difficulties, which often arise through lack of clear definitions, when comparing different systems.

In this section I will describe the advantages and disadvantages or the reasoning behind each option and will justify the choices I have made when defining the problem I am trying to solve.

2.2.1. Music and emotion

For the last twenty years or so, interest in emotion and music has been growing, and it is attracting attention from a wide range of disciplines: philosophy, psychology, sociology, musicology, neurobiology, anthropology and computer science.

After early debate about whether or not music could express or induce emotions at all, both are now generally accepted with multi-disciplinary backing. Peretz [2010] has shown that emotion in music is shared between different cultures, and is therefore universal and related to the basic set of emotions in people. It also has as strong an effect as everyday emotions, activating the same or similar areas in the brain (see Koelsch et al. [2010]).

As the general field of musical emotion is quite a bit older than computational emotion recognition in music, there has now accumulated a large set of studies into the effects that different musical features have on the emotion in music. A summary of such a set can be seen in the chapter by Gabrielsson and Lindström [2010] with a summary of that set shown in Table 2.1. Most of the polar values of musical features appear on either the opposite ends of a single axis (arousal or valence) or along the diagonal between the two axes. As can be seen from the summary, there is a similar distribution of effects on both the arousal and the valence axes. While some of these findings are used in computational emotion recognition in music, a lot of the musical features or their levels are difficult to extract from just the wave representation of a song, and so are often replaced by much lower level features. We have seen a similar approach being taken in the field of voice emotion recognition (Section 2.1.3), and some of the low-level features used in music emotion recognition (MER) are actually borrowed from the field of speech recognition (e.g. MFCC).

2.2.2. Type of emotion

There are two types of musical emotion one can investigate—emotion "expressed" by the music (or the perceived emotion), and emotion induced in the listener (or the felt emotion). The former is concerned with what the music sounds like and is mainly influenced by the musical features and cultural understanding of music. It is also more objective, since the listener has little or no effect on the emotion. The latter, on the other hand, describes the listener's response to a piece of music. It clearly depends on the perceived emotion (expressed by the music), but is also heavily influenced by the individual's experiences, history, personality, preferences and social context. It is therefore much more subjective and varies more between people.

Even thought the vast majority of papers in MER do not make the distinction, there is clear evidence that the two are different. In their study, Zentner et al. [2008] found a statistically significant difference between the (reported) felt and perceived emotions in people's reported emotional response to music. They also found that certain emotions are more frequently perceived than felt in response to music (particularly the negative ones), and some are more frequently felt rather than perceived (e.g. amazement, activation, etc.).

It has also been suggested by Gabrielsson [2002] that there can be four different types of interactions between the felt and perceived emotions: positive relation (e.g. happy music inducing happy emotion), negative relation (e.g. sad music inducing happy emotion), no systematic relation (e.g. happy music not inducing any emotion) and no relation (e.g. neutral music inducing happy emotion). This theory has also been confirmed by Kallinen and Ravaja [2006] and Evans and Schubert [2008]. Both studies found that felt and perceived emotions are highly correlated, but that there is also a statistically significant difference between them.

While initially the datasets of emotion annotations either did not indicate which emotion type they are referring to or used perceived emotion, most of the datasets nowadays (and especially the ones used in my work) tend to explicitly ask for perceived emotion labels from their participants. Despite the fact that we know that people distinguish between the two types of emotions and give different responses, great care must be taken to explain exactly what is being asked. Unfortunately, instructions given to the participants in online studies often only briefly mention the type of emotion of interest, or use unnecessarily complex language to express the goal of the study. It is not clear how carefully participants of online trials read the instructions and we can therefore only assume that the law of averages means that the results we get are representative of the expressed emotion in music.

2.2.3. Emotion representation

As discussed in section 2.1.1, there are different ways in which emotion can be represented. In MER, only the dimensional and categorical approaches are used, and the appraisal model is largely ignored. Historically, the categorical representation was much more common, but recently this trend has started to change—a higher

and higher proportion of papers published each year use the dimensional rather than the categorical models.

Over the last 10 years, researchers have come up with numerous different approaches to categorical emotion representation. Feng et al. [2003a,b] used 4 mood classes (happiness, sadness, anger and fear) which were based on work by Juslin [2000] on music perception. Li and Ogihara [2003] based their emotion labels on the Farnsworth [1958] theory that groups adjectives used to describe music into ten groups, but then modified them by allowing their participant to add his or her own labels.

Another approach is to try to mine various databases using statistical tools in order to extract a set of labels that might be more natural and universal when talking about emotion in music. Hu and Downie [2007] derived their five mood clusters (rousing, cheerful, wistful, humorous, aggressive/intense) from an online music-information service `AllMusicGuide.com`, by running a clustering algorithm on a similarity matrix of the 40 most popular mood labels used on the site. These mood clusters were later adopted by the MIREX audio mood classification task [Hu et al., 2008; Tzanetakis, 2007]. Skowronek et al. [2007] selected 12 categories from 33 candidate labels used in the literature based on how important and easy to use they are, while Zentner et al. [2008] extracted the ten most discriminating musical mood labels after having a large number of participants rate a list of 515 affect labels.

A fairly common approach is one that lies in between the categorical and dimensional approach: basing the categorical labels on the dimensional model. Most commonly, only the four labels that relate to the four quadrants are used (exuberance, depression, contentment, anxiousness) (used by Liu et al. [2003]; Yang et al. [2006]; Liu [2006]). There are also studies that include labels that relate to the axes too—Wang et al. [2004] added "robust" and "sober" that imply valence being neutral; and studies that use more than two dimensions (e.g. that by Trohidis and Kalliris [2008]).

With the increasing popularity of the dimensional classification, there have been investigations of the appropriate set of axes for a music emotion space. It has repeatedly been shown that adding a third (dominance, tension, etc.) axis has little or no discriminative power between different emotions in music, as it correlates highly with the arousal (or power) axis (see Eerola and Vuoskoski [2010]; MacDorman and Ho [2007]; Eerola et al. [2009]; Evans and Schubert [2008]). It is interesting to note that similar results have been found in other fields of affective computing. For example, the dictionary of affective norms collected by Warriner et al. [2013] (described in Section 2.1.4) found a linear correlation between the valence and dominance axes. This suggests that, at least for sentiment analysis in text, the dominance axis might not be of much use. In addition to that, Evans and Schubert [2008] and MacDorman and Ho [2007] reported that participants found the addition of the third axis confusing and difficult to deal with. Given these findings, it is not a surprise that a majority of researchers [Schubert, 2004; Yang et al., 2006; Han et al., 2009; Schuller et al., 2011; Huq et al., 2010; Kim, 2008] use the arousal and valence axes only. Eerola and Vuoskoski [2010] have also shown

that dimensional representation can be more reliable than a categorical approach, especially for emotionally ambiguous pieces, since it provides higher resolution of labels. Schmidt et al. [2010] reached the same conclusion by showing that regression achieved better results compared to AV based 4-class classification, especially in cases where the labels fall close to an axis and a small error in prediction can lead to complete misclassification of a piece.

2.2.4. Source of features

The vast majority of MER research has been done using only the acoustic features extracted from the whole audio signal. Since high-level features can be difficult to extract accurately, the work is mostly based on low level features such as mel-frequency cepstral coefficients (a representation of short-term power spectrum, widely used in voice related research), chroma, loudness, spectral centroid, spectral flux, rolloff and zero crossing rate. Tempo, rhythm, mode, pitch, key, chord are also used, but much more rarely, since they can be hard to extract in general, and especially in case of music which potentially contains more than one melody line. There is also difficulty in extracting perceived (high-level) features, given that there are no clear rules how or why we e.g. prioritize one rhythm and ignore others in a song. See Section 4.1 for a fuller description of lower-level features and the techniques used for extracting them.

Despite these shortcomings, low-level acoustic features have been very useful in MER, especially for the arousal axis in dimensional representation of emotion. With these features alone, the R^2 statistic for regression models can reach around 0.7 for arousal, but only 0.3 for valence.

To address this gap in results, researchers started investigating the effect that lyrics have on emotion in music. As discussed in section 2.1.4, a large proportion of sentiment analysis in text is concerned with positive/negative labels, which relates well to the valence axis in the dimensional model. And it seems to be the case in lyrics too—Hu et al. [2009a], Yang et al. [2008] and others have found that features extracted from lyrics are better at explaining variance in valence than in arousal.

Most work that relies on lyrics for music emotion recognition falls into one of three categories—they either use TF-IDF (or similar) weighted vector space models [Yang et al., 2008; Mahedero et al., 2005; Laurier et al., 2008], n-gram models [He et al., 2008; Hu and Downie, 2010a,b] or knowledge-based models [Yang and Lee, 2004; Hu et al., 2009b; Xia et al., 2008]. See Section 6.2.1 for a more detailed description of some of these techniques.

Those that chose to combine features extracted from lyrics and acoustic features [Yang et al., 2008; Hu and Downie, 2010a,b; Laurier et al., 2008; Schuller et al., 2011, 2010] have found that the hybrid model always outperforms models that are based on either textual or audio features alone. This confirms the findings from psychological studies [Ali and Peynircioglu, 2006; Besson et al., 1998] that have shown the independence of lyrics and music and highlights the need of multimodal approaches to MER.

2.2.5. Granularity of labels

For both the categorical and the dimensional emotion representations, there is an important question that we need to answer when we are building a MER system—how many labels are we going to allow for each song? There is absolutely no doubt that emotion in music can change (whether it is only expressed or induced), therefore restricting the user to only one category or one point in the affect space can seem limiting and lead to errors both in labeling and in emotion recognition. Most of the work in MER so far has been done on static categorical emotion representation, but the trend is changing towards dynamic dimensional representation.

Studies using categorical emotion representation tried to solve this problem by allowing multiple labels to be chosen for each song. Li and Ogihara [2003] used a set of binary classifiers and posed no limitations on their participants in how many labels to choose. Trohidis and Kalliris [2008] compared several training models that approach the problem of multi-label classification differently—ranging from binary classification, to label supersets as classes, to models that are adapted to the problem of multi-label classification (showing the best performance). Another way of dealing with this problem was introduced by Yang et al. [2006]—they suggested using a fuzzy classifier that infers the strength of each label.

The dimensional representation offers a completely different solution—time varying MER. Even though it is clearly not restricted to the dimensional approach (as has been shown by Liu [2006], who suggested that the music piece can be automatically segmented into segments of stable emotional content and apply static emotion recognition on them, and Schubert et al. [2012], who introduced an approach for continuous categorical emotion tagging), the time varying categorical representation is inherently more difficult to use, especially in user studies.

Even within dimensional emotion tracking, there are different ways of approaching the problem. Korhonen et al. [2006], Panda and Paiva [2011], Schmidt and Kim [2010a], Schmidt et al. [2010], and others have tried to infer the emotion label over an individual short time window. Another solution is to incorporate temporal information in the feature vector either by using features extracted over varying window length for each second/sample [Schubert, 2004], or by using machine learning techniques that are adapted for sequential learning (e.g. sequential stacking algorithm used by Carvalho and Chao [2005], Kalman filering or Conditional Random Fields used by Schmidt and Kim [2010b, 2011a]). Interestingly, it has also been reported by Schmidt et al. [2010] and Panda and Paiva [2011] that taking the average of the time-varying emotion produces results that are statistically significantly better than simply performing emotion recognition on the whole piece of music.

2.2.6. Target user

The last important choice one has to make when designing an MER system is whether it is designed to be personalized or universal. As suggested by Chai and Vercoe [2000], user modelling can be used to reduce the search space in information retrieval systems, improve the accuracy of the systems, etc. They identify two

groups of features that could be used in a music information retrieval system—quantitative (derived from the music, not differing much from person to person) and qualitative (entirely user dependent).

Yang et al. [2007] used this idea to build a system that compared a baseline regressor (trained on average opinions only), group-wise regressor (where users were grouped by their sex, academic background, music experience and personality) and a personalized regressor (trained on explicit annotations by a particular user) in combination with the baseline regressor. The personalized approaches gave a statistically significant improvement over the baseline method, but require substantial user input. The group-regressor, on the other hand, showed no statistical improvement over the baseline, which suggests that the benefits of personalized approach cannot be achieved with simple grouping techniques.

Another system that was based on the idea of personalizing MER was proposed by Yang and Lee [2009] and used a collection of models trained on individual user to train a "super regressor". It was then used to model the perception residual of a user. The combination of the two showed a significant improvement over a baseline model that was trained on the average opinion only.

Since emotion recognition in music is bound to be subjective (especially if we consider the induced emotion), it is not surprising that personalizing the systems brings an improvement. Although given that the accuracy (especially for the valence axis) was not perfect, it still means that we need a strong underlying universal model, especially if users are unwilling to pay the price of providing the input required to personalize the system.

2.2.7. Continuous dimensional music emotion recognition

For the reasons described in Sections 2.2.2-2.2.6 I have focused my work on the dimensional continuous perceived emotion recognition for a general user. Such a model is the most general and can then be used as a subsystem for other models, if necessary.

There has already been some work done on this problem. Given the large differences in datasets, sometimes unclear emotion types (see Section 2.2.2) and approaches to system evaluation (see Section 3.1) used, it is hard to compare the results achieved by different studies. The correlation achieved by different systems usually fall within the same range—with arousal values much higher than valence. R^2 statistics for valence axis usually fluctuate between 0.2 and 0.4, while R^2 for arousal usually falls between 0.6 and 0.8.

Most, if not all, approaches to continuous dimensional emotion recognition focus on acoustic analysis based machine learning models. While some work has been done on the development of features that are especially suitable for the task (e.g. work by Schmidt and Kim [2011b] on deep learning approach to building a feature vector or work by Kumar et al. [2014] on designing new audio features that correlate well with arousal and valence of a song), most research is focused on building new, more appropriate machine learning models.

Support Vector Regression (SVR) is a commonly used model when the focus is not the actual machine learning model itself. For example, Schmidt and Kim [2010a] used it to build a hierarchical model where regressors trained on a single type of feature are combined by another regressor. Scott et al. [2012] also used a hierarchical model, except in their work they had a separate linear regressor for each individual channel of a song (vocals, piano, bass and drums).

In addition to SVR and linear regressors, there has now been a wide range of more advanced machine learning models applied to the problem. Wang et al. [2012] introduced the acoustic emotion Gaussians model which models the distribution of emotion labels instead of trying to predict a particular point on the AV space. A similar approach was suggested by Schmidt and Kim [2011a] using conditional random fields and quantizing the AV space into an 11x11 grid of squares and representing the prediction as a heatmap on this space. Other models, such as Long-Short Term Memory Recurrent Neural Networks by Coutinho et al. [2014], allow the encoding of temporal information, which a lot of the other models lack.

2.2.8. Displaying results

Dimensional continuous emotion recognition introduces another problem to the field of music emotion recognition—display of 3-dimensional results. While there is an easy and intuitive way to represent the results of emotion recognition for the whole song in both the categorical and dimensional emotion representation, having a static representation of 2-dimensional emotion labels on a time-scale is not straightforward.

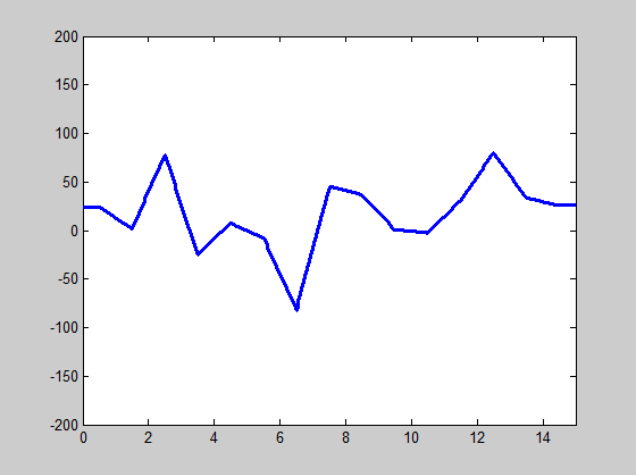


Figure 2.3: An example of a one dimensional representation of emotion prediction over time, where time is shown on the horizontal axis, and the dimension in question is represented by the vertical axis

If we are working with one axis at a time, or even with continuous categorical labels, it is easy to display the results on a 2-dimensional graph with x-axis usually representing time, and y-axis representing either the dimensional axis of interest

or the confidence we have in a particular categorical label (see Figure 2.3). It is also easy, in such a case, to visually compare the prediction with the ground truth.

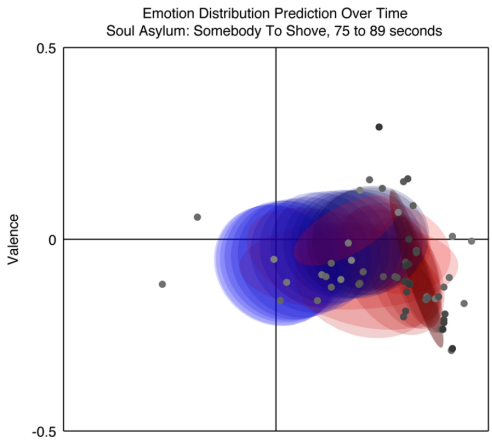


Figure 2.4: An example of a two dimensional representation of emotion prediction over time (valence on the vertical axis and arousal on the horizontal axis), taken from Schmidt et al. [2010], where time is represented by the darkness of the dots (labels), or the color of the ellipses (distributions)

Where it all falls apart is when we add another axis. One solution to this problem is to separate the axes out and have a graph for each one of them. Unfortunately, this makes it a lot more difficult to notice global trends and any interaction that might happen between the 2 axes we are interested in. A common approach (used by Schmidt et al. [2010]) is to colour-code or size-code time instead of representing it with a separate axis (see Figure 2.4, where time is encoded by both the darkness of the dots or the projected eclipses to represent distribution)—the dot starts off being of a particular size or colour and it changes as time goes by. That also works well as a video, which can then be represented by a static image—I have used this approach in my evaluation metrics study (see Chapter 3). Another solution proposed by Cowie et al. [2009] is to extend the 1-axis approach by also incorporating colour into it (see Image 2.5). The y-axis still represents the intensity of the emotion, but here the intensity is defined as the distance form the origin to the point in the AV space. The colour is used to encode the location of the emotion in the AV space—it is the product of interpolation between red-green valence axis (red representing the negative end and green–the positive end of the axis) and the yellow-blue arousal axis (yellow representing the active end and blue–the passive end of the axis).

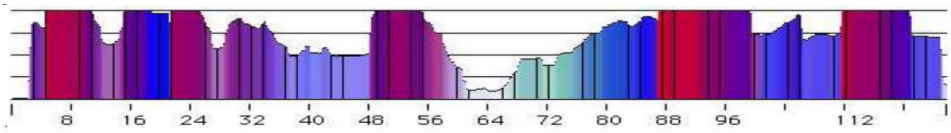


Figure 2.5: An example of a 2.5 dimensional representation of emotion prediction over time (height represents the intensity of emotion and colour is an interpolation between red-green axis of valence and yellow-blue axis of arousal with time on the horizontal axis), taken from Cowie et al. [2009]

Factor	Effect on emotion	Levels
Amplitude envelope	Low arousal and valence vs. high arousal and valence	Round, sharp
Articulation	High vs. low arousal	Staccato, Legato
Harmony	High valence and low arousal vs. low valence and high arousal	Simple/consonant, complex/dissonant
Intervals	High vs. low valence; high valence and arousal vs. low valence and arousal Various effects on both axes	Harmonic: consonant, dissonant; high-pitched, low-pitched Melodic: large, minor 2nd, perfect 4th, major 6th, etc.
Loudness	High vs. low arousal	Loud, soft
Loudness variation	Low valence vs. high valence (in both)	Large, small; few/no changes, rapid changes
Melodic (pitch) range	High vs. low arousal	Wide, narrow
Melodic direction	Mixed	Ascending, descending
Pitch contour	High vs. low arousal	Up, down
Distribution of intervals in melodies	High arousal; high valence	Minor seconds and intervals larger than the octave, unisons and octaves; perfect fourths and minor sevenths
Mode	High vs. low valence	Major, minor
Pause/rest	Low vs. high arousal	After tonal closure, after no tonal closure
Pitch level	Mixed, but mainly high valence and arousal vs. low valence and arousal	High, low
Pitch variation	High valence vs. low valence	Large, small
Rhythm	Low arousal vs. high arousal; low vs. high valence	Regular/smooth, irregular/rough; complex or firm, varied or flowing/fluent
Tempo	High vs. low arousal	Fast, slow
Timbre	Low vs. high arousal	Few harmonics or soft; many harmonics or sharp
Tonality	High vs. low valence	Tonal, atonal or chromatic

Table 2.1: Suggested set of musical features that affect emotion in music (adapted from Gabrielsson and Lindström [2010])

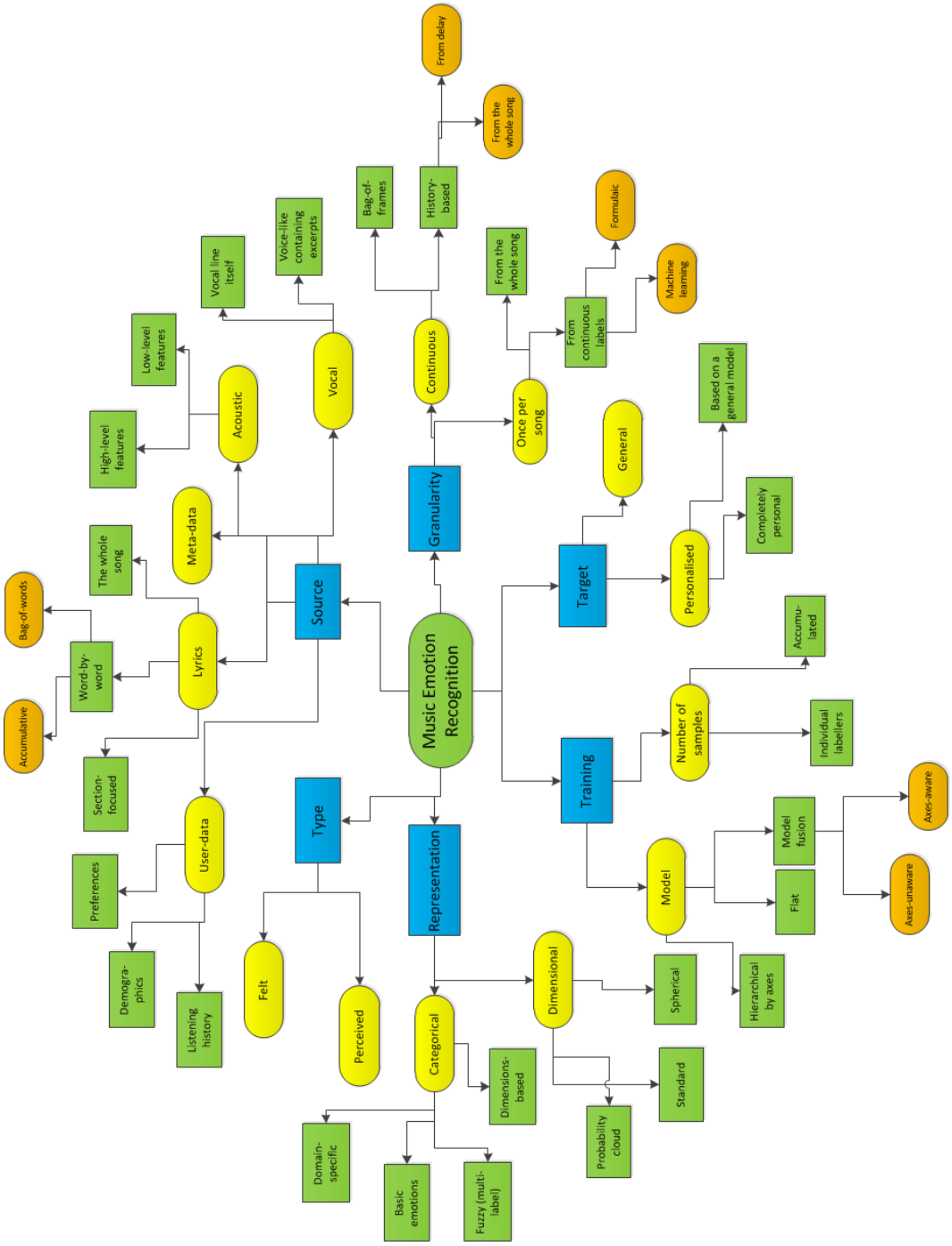


Figure 2.2: Summary of the various choices that need to be made for emotion recognition in music

2.3. Music emotion recognition challenges

While setting up guidelines and agreeing on experimental design is one way to improve the comparability of work done by different researchers, there is another approach that allows a direct comparison between different approaches—through challenges. In these, researchers are required to submit their code or their solution to a particular well defined problem. The development sets are usually shared, and the test set and the evaluation is kept identical for all the participants allowing a ranking of various techniques to be produced.

The first emotion recognition in music challenge was the Music Mood task in the MIREX challenges organized by Hu et al. [2008], and set up as part of the International Conference on Music Information Retrieval—the MIREX challenges have been organized since 2005, although the Music Mood task was only introduced in 2007. In this task, the participants were encouraged to submit solutions to a music emotion classification problem—where a single label (category) was predicted for a whole song. The task has been running ever since, using the same dataset of 600 songs, 120 in each of the 5 mood clusters. The 5 mood clusters were derived by Hu et al. [2008] from the AllMusicGuide[3] metadata collection using hierarchical clustering. The dataset covers a large set of 27 genres and contains 30 second annotated extracts. While the improvement in the results was quite large at the beginning, as time went by, the difference between the participants was getting smaller, as well as the improvement compared to previous years.

As the continuous, dimensional emotion tracking in music was introduced later than the emotion classification of the whole song, so were the challenges for researchers trying to solve this problem. The first attempt was the MediaEval 2013 Emotion in Music task organised by Soleymani et al. [2013a] that required the participants to provide both the continuous dimensional labels, and the dimensional labels for the whole extracts. While the task attracted only 3 teams, the task was considered important enough to be repeated the following year. In 2014 the Emotion in Music task in MediaEval changed slightly—the training and testing sets were made larger, and the granularity of the labels was increased (from 1Hz to 2Hz) (see Aljanaki et al. [2014]). The task had also dropped the overall static labels for the whole extract, hinting that if we can do a time-based emotion labelling, then that is what we should focus on. The task attracted more participants (it had 6 teams) with a wider set of approaches: multi-level regression by Fan and Xu [2014], Long-Short-Term-Memory Recurrent Neural Networks by Coutinho et al. [2014], State-Space Models by Markov [2014], etc. The datasets used in both tasks are now publicly available for the researchers to use.

In addition to creating public datasets—valuable in a field so dependent on training and testing data—such tasks allow a direct comparison between different approaches. In a field that does not have a set of agreed standards and where small (or big) differences in evaluation techniques can make the results impossible to compare, this gives an invaluable opportunity to not only compare the work, but to also discuss how such comparison should be made.

[3] http://www.allmusic.com/

2.4. Datasets

An important part of a good solution to a machine learning problem is a good dataset. A publicly available dataset is also a necessary requirement for a research solution, so that the work could be compared to that of others. As the field of continuous dimensional emotion recognition is new, public labelled datasets are hard to find. In addition to the fact that labelling such datasets is a labour- and time-intensive task, another problem is the copyrights of the songs—as researchers start focusing on popular music, as opposed to classical pieces, the issue of sharing the audio files in addition to the annotations becomes problematic.

Most of the datasets available now are annotated online, and so the labels are expected to be noisy. The saving grace of such an approach is that it is much easier and faster to collect annotations, and we can therefore have more people label each song. It is interesting to note that despite the fact that the collection forms that are used in different annotations are fairly different, the average SD of a label between all the participants for each song is around 0.3 (where the range of each label is from -1 to 1) for both axes in all the datasets (see their descriptions below). Comparing these datasets with affective norms dictionaries (0.4 SD for valence and 0.5-0.6 for arousal, based on the same range) that are also annotated using an online service (see Section 2.1.4) we see that we actually get a smaller SD of labels in music annotations. While the difference in SD for arousal and valence is expected (and present in the affective norms dictionaries), it does not seem to appear in song annotations, despite the large difference in the results achieved by models for the two axes.

2.4.1. MoodSwings

The MoodSwings Turk dataset is probably the first publicly available dataset that was designed for continuous dimensional emotion recognition in popular music—it is therefore the main dataset used in this dissertation. The music extracts are labeled on the arousal-valence dimensional space, where the annotations have been sampled at 1Hz sampling rate. The data has been collected by Speck et al. [2011] using MTurk, asking paid participants to label 15-second long excerpts with continuous emotion ratings on the AV space, with another 15 seconds preceding those given as a practice for each song. The dataset consists of annotations for 240 15-second extracts (without the practice run) with on average 16.9 $\pm$ 2.7 ratings for each clip (examples of which can be seen in Figure 2.6). The songs used in the dataset cover a wide range of genres—pop, various types of rock, hip-hop/rap, etc, and are drawn from the "uspop2002"[4] database containing low-level features extracted from popular songs, but not the songs themselves (due to copyright issues). In addition to the annotations, the dataset contains a standard set of features extracted from those musical clips: MFCCs, octave-based spectral contrast, statistical spectrum descriptors, chromagram and a set of EchoNest [5] features. EchoNest is a commercial API that provides access to a large array of data related to mil-

[4] http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html

[5] http://developer.echonest.com/downloads

lions of songs. It is easy to use and can be quite useful, but as it is a commercial product, it is not clear how those features are extracted, and the reproducibility of the results based on EchoNest features is limited.

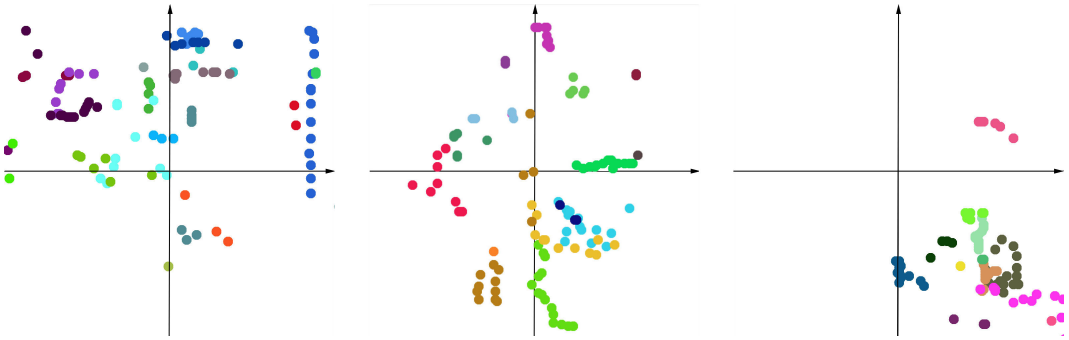


Figure 2.6: Labeling examples for songs (from left to right) "Collective Soul - Wasting Time", "Chicago - Gently I'll Wake You" and "Paula Abdul - Opposites Attract". Each color corresponds to a different labeller labelling the same excerpt of a song

Kim [2008] first attempted to collect the same data using a collaborative online game, designed to make the annotations fun, competitive, but also encouraging consensus. The design of the game was deemed successful, but due to a lack of ongoing player interest and potential bias in the annotations, the game was replaced by MTurk as a way to collect training and testing data. MTurk can provide fast and cheap way of collecting a large amount of data, but it raises worker trust issues, lacks the necessary control and introduces the need to filter the responses, which can be hard to do automatically.

While for most of my work, the features provided in the dataset would have been sufficient, there were several experiments that required the actual audio files (see Chapter 6). For these I attempted to acquire the same recordings that were used in the data collection study. I was able to get access to 203 songs that matched the artist and the album used in the study, which were then used for the experiments that involved feature extraction, discarding the annotations for the remaining 37 songs. When the full dataset containing 240 songs is used, it is referred to in this dissertation as the original MoodSwings dataset.

2.4.2. MediaEval 2013

As part of the Emotion in Music task in MediaEval 2013 [Soleymani et al., 2013a], a new dataset was developed to be used as the development and the testing set for the challenge. Soleymani et al. [2013b] collected 1000 songs from the Free Music Archive (FMA)[6], covering a wide range of western music genres (blues, jazz, classical music, pop, rock, electronic music, etc.). 700 of those songs were designed to be used as the development set, and the remaining 300 as the test set. 45 second clips were extracted from the songs at random locations within each audio, and the clips were annotated using the MTurk service at 1Hz sampling rate on the Arousal-Valence space. The first 15 s of annotations for each song were discarded

[6] http://freemusicarchive.org/

as a practice run. The majority of songs on the FMA are under the Creative Commons license, which means that the actual audio clips can be distributed together with the annotations, making this dataset especially useful for research in emotion in music. The audio clips provided are re-encoded to have the same 44100 Hz sampling rate.

2.4.3. MediaEval 2014

Another dataset that features in this dissertation is the dataset used for the Emotion in Music task in the MediaEval 2014 challenge, developed by Aljanaki et al. [2014]. The overall structure of this dataset is similar to that of the Emotion in Music task from MediaEval 2013. It is a publicly available development dataset of 744 songs that were selected from the MediaEval 2013 task of Emotion in Music—after the task, the organizers identified a set of duplicates, therefore reducing the initial set from 1000 songs to 744. The same 45 second clips were extracted from those songs with the emotion labels discarded for the first 15 seconds and provided for the subsequent 30 seconds. Unlike in the previous task, the sampling rate of the emotion labels was doubled to 2 Hz from the 1 Hz used in the previous task the MoodSwings dataset. Similarly to the other two datasets, the song annotations were done online, using the MTurk service. The testset contains an additional 1000 songs from FMA annotated in the exact same way as the development set. The songs in the dataset are still only focusing on mainstream Western music, covering a variety of genres from pop to classical music.

Evaluation metrics

3

Continuous emotion representation in the arousal-valence space is widely used in affective computing. Unfortunately, as is often the case with new disciplines, so far there is a noticeable lack of agreed guidelines for conducting experiments and evaluating algorithms. In addition, many datasets are used, which have been collected in different ways and for different purposes. As researchers make design choices that are appropriate for their particular investigation and that differ from the choices made by other researchers, comparing their work is difficult. What makes it even worse is that choosing a different evaluation metric can change the order in which various methods would be ranked, making the correct choice of an evaluation metric vital.

To address this issue, I decided to investigate the differences and similarities between different evaluation metrics. As that gave little insight into which evaluation metric is the most appropriate, I designed and executed a study to find out how people perceive the "goodness" of different evaluation metrics. By identifying which evaluation metric matches people's intuition the best, I suggest that using the same metric for algorithm optimization will lead to results that will be closer to people's perception of emotion in music. As the ground truth is based on the opinion of an average user, the output of a classification algorithm should be too.

The work described in this chapter has been done in collaboration with Tadas Baltrušaitis who helped with MATLAB scripts, advised on the design of the study and helped running it. The work has been published in:

What really matters? A study into people's instinctive evaluation metrics for continuous emotion prediction in music, Vaiva Imbrasaitė, Tadas Baltrušaitis, Peter Robinson, *Affective Computing and Intelligent Interaction, Geneva, Switzerland, September 2013*

3.1. Evaluation metrics in use

Each field that deals with emotion recognition or sentiment analysis tends to use different evaluation techniques. In the cases where classification is the main tool used, there is generally more consensus, thanks to the influence of the field of Information Retrieval. When regression or other, graphical, machine learning meth-

ods are employed, a number of different techniques get used. A representative sample of them is summarized in Table 3.1. There is a clear grouping that can be seen based on the authors of the papers—unsurprisingly, the same people tend to use the same evaluation techniques for their algorithms. Unfortunately, as is also clear, there is little overlap between the metrics used by different research groups. In order to achieve a somewhat less biased view of the field, the summary excludes any papers published by me or my research group. I would also like to note that this table gives a slightly more optimistic view of the research as it groups together MSE and RMSE, as well as all the different kind of correlation coefficients that get used.

It is encouraging to see that the new Emotion in Music tasks are using the metrics that I am advocating based on the results of this chapter. It is interesting to note that they are in the minority of groups in the music emotion recognition field (first half of the table) to use short metrics (see Section 3.1.7 for the explanation of the difference between a long metric and a short one), while it is much more common in the field of emotion recognition from video (second half of the table). It is probably not too surprising that this is the case since the organisers of the task come from the general field of emotion recognition and short metrics are used there more often than in the field of music.

Table 3.1: Summary of metrics used for evaluation of continuous emotion prediction—music research at the top and facial expressions at the bottom. Starred entries indicate that the sequence length used in the paper was not made clear and the entry in the table is speculated

	Long MSE	Long corr.	Variance of param.	KL divergence	Euclidean distance	Earth Mover's distance	"Pixel" error	Short MSE	Short corr.	Kendall's τ	Sign agreement
Korhonen et al. [2006]	x	x	x								
Schmidt and Kim [2010b]				x	x						
Schmidt and Kim [2010a]				x	x						
Schmidt et al. [2010]					x						
Schmidt and Kim [2011a]					x	x	x				
Scott et al. [2012]				x	x						
Wang et al. [2012]				x	x						
Weninger and Eyben [2013]					x				x	x	
Emotion in music in MediaEval 2013/2014 (Section 2.3)								x	x		
Kanluan et al. [2008]		x*			x						
Grimm and Kroschel [2007]					x				x		
Wöllmer et al. [2008]	x*										
Nicolaou et al. [2011]								x	x		x
Nicolaou et al. [2012]								x			

The use of different evaluation metrics can without a doubt make it difficult to compare the work done by different researchers. What might not be immediately obvious is that the choice of a different metric can influence the ranking of the different methods used. A good example of that is my submission to the MediaEval 2014 Emotion in Music task. A quick look at Table 5.10 will immediately make it obvious that the choice between correlation and RMSE, for example, in that particular example would completely change the order in which the 4 methods that were used should be ranked.

3.1.1. Emotion in music

Even though the majority of work in the field of emotion recognition in music is done on emotion classification, there is already a significant body of research done on continuous emotion prediction in the dimensional space (using regression).

When classification is used, the confusion matrix is often included in the analysis of the results, giving a clear image of, and quite a lot of insight into, the model. The most common evaluation metric used is accuracy or classification error, which are occasionally replaced by more complex metrics such as F1 score.

Within the research on dimensional musical emotion prediction, there is a wide range of evaluation metrics used. Starting with the standard metrics such as RMSE and correlation—which are often calculated and reported for each dimension separately—[Korhonen et al., 2006], but also including Kullback-Leibler divergence [Schmidt and Kim, 2010b,a], average Euclidean distance [Schmidt et al., 2010; Schmidt and Kim, 2011a] and Earth mover's distance [Schmidt and Kim, 2011a]. Historically, these metrics were calculated by first concatenating all the audio extracts into one and then using at as a single sequence, but recently there have been examples where the average of per-song metrics is reported. The evaluation sections of many papers often lack detail, failing to specify how the sequence is defined and which exact form of a metric is used.

3.1.2. Emotion recognition from audio/visual clues

The idea of modeling emotion in terms of several latent dimensions is not exclusive to music. Such representation of affect is used when modeling external expressions of emotions such as emotional speech, facial expressions, head gestures, and body posture.

The types of metrics used to evaluate automatic prediction of affect have been varied. First of all it depends if the task is framed as a classification or regression one. For the classification tasks (such as in Audio/Visual Emotion Challenge 2011) the metrics used are accuracy (equivalent to sign agreement if regression values are turned into classes based on the axes used) and F1 scores. Sometimes the classification is formed as a 4 or 8 way problem instead of just a binary one as well—then F1 scores are used, or confusion matrices are presented. Furthermore, some work has been done with splitting it all into dense quantised levels (i.e. 5 or 7 levels) and classifying them.

When the problem is formed in continuous space (such as Audio/Visual Emotion Challenge 2012 [Schuller et al., 2012] and 2013[1]) metrics such as average RMSE [Nicolaou et al., 2012; Gunes et al., 2011]), correlation and sign-agreement [Gunes et al., 2011] per sequence are used. The metrics are usually reported per dimension (separate scores for valence, arousal etc.). Unfortunately, many authors fail to make it clear whether the evaluation metrics they report are averaged across sequences or computed from a single concatenated sequence (as is more common in music community), making it more difficult to compare different work.

3.1.3. Emotion recognition from physiological cues

Emotion recognition based on the analysis of physiological measurements could provide a way of evaluating the felt emotion, as opposed to the expressed emotion (or the mixture of both). There is a variety of measurements that such a system could be based on: EKG, skin conductivity, heart-rate variability, EEG, etc. Classifiers instead of regressors are often used, with accuracy as the evaluation metric [Calvo and D'Mello, 2010]. Even when regressors are used initially, the final outcome is commonly converted to a class by using a set of bins for the labels and accuracy as the evaluation metric [Lotte et al., 2007].

3.1.4. Sentiment analysis in text

In the field of sentiment analysis in text, the majority of work tends to focus on the valence axis only [Pang and Lee, 2008]. Even though the task of inferring how positive a piece of text is would lend itself naturally to regression, it is often approached as or converted to a classification problem, binary or ordinal [Pang and Lee, 2005; Mao and Lebanon, 2007]. In the case of classification, accuracy is used as the evaluation metric, sometimes with the addition of RMSE [Wilson et al., 2004]. For tasks defined as regression, correlation is used [Bestgen, 1994; Dzogang et al., 2010].

3.1.5. Mathematical definition

One dimensional case

The simplest approach to emotion recognition is to consider each affective attribute as a separate dimension. As seen in the background section (Section 3.1) there are a multitude of metrics used to evaluate the machine learning algorithms for the task of dimensional emotion prediction. If we consider a sequence of length n with a ground truth $g(x)$ and prediction $p(x)$ per time-step x, we can define the most common metrics used in the field.

Average Euclidean distance:

$$E_{\text{Eucl}}(g,p) = \frac{1}{n}\sum_{i=1}^{n} ||g(i) - p(i)|| \tag{3.1}$$

[1] http://sspnet.eu/avec2013/

Root mean square error (RMSE):

$$E_{\text{RMSE}}(g,p) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(g(i)-p(i))^2} \tag{3.2}$$

Pearson correlation coefficient:

$$E_{\text{Corr}}(g,p) = \frac{\sum_{i=1}^{n}[(g(i)-\overline{g})(p(i)-\overline{p})]}{\sqrt{\sum_{i=1}^{n}(g(i)-\overline{g})^2}\sqrt{\sum_{i=1}^{n}(p(i)-\overline{p})^2}}, \tag{3.3}$$

where $\overline{g}$ and $\overline{p}$ are the mean ground truth and predictor values for the sequence of interest. Some authors use squared correlation coefficients instead of non-squared ones, I choose not to do so. Squaring the correlation coefficient can hide the fact that the predictions are inversely correlated with ground truth, which is not a desired behaviour of a predictor. It is especially misleading when the average per-song correlation is used—if only some of the predictions are inversely correlated, the standard argument of inversing the prediction no longer applies and in such a case the squared correlation would hide the inconsistent and poor behaviour of a model.

I use the definition of the **average sign agreement (SAGR)** from Gunes et al. [2011]:

$$E_{\text{SAGR}}(g,p) = \frac{1}{n}\sum_{i=1}^{n} s(g(i),p(i)), \tag{3.4}$$

$$s(x,y) = \begin{cases} 1, & \text{sign}(x) = \text{sign}(y) \\ 0, & \text{sign}(y) \neq \text{sign}(y) \end{cases} \tag{3.5}$$

I also define the average per frame **Kullback-Leibler divergence (KL-divergence)** for Normal distributions. KL-divergence is a metric that measures the difference between two probability distributions, and is often suitable for the task at hand.

$$E_{\text{Mean-KL}}(g,p) = \frac{1}{n}\sum_{i=1}^{n} E_{\text{KL}}(p(i),g(i),\sigma_{p(i)},\sigma_{g(i)}) \tag{3.6}$$

$$E_{\text{KL}}(p,g,\sigma_p,\sigma_g) = \frac{1}{2}(\sigma_g^{-1}\sigma_p + (p-g)\sigma_g^{-1}(p-g) - 1 - \ln(\sigma_g^{-1}\sigma_p)) \tag{3.7}$$

The predictor could provide an estimate together with uncertainty (σ_p) and the ground truth can be modeled as a Normal distribution as well (centered on mean with σ_g calculated from the labels from multiple people). It is, however, much more commonly used with multiple dimensions.

Two dimensional case

A stronger approach to two-dimensional models is to consider two affective attributes (typically valence and arousal) simultaneously. We have two predictors p_1 and p_2 (or a single non-correlated predictor for both dimensions $\mathbf{p}$), we also have the ground truth g_1 and g_2 for both dimensions.

Average Euclidean distance:

$$E_{\text{Eucl}} = \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{g}(i) - \mathbf{p}(i)|| \tag{3.8}$$

Root mean square error (RMSE):

$$E_{\text{RMSE}}(g, p) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} ((g_1(i) - p_1(i))^2 + (g_2(i) - p_2(i))^2)} \tag{3.9}$$

Average correlation across dimensions:

$$E_{\text{Corr}}(g, p) = \frac{1}{2}(E_{\text{Corr}}(g_1, p_1) + E_{\text{Corr}}(g_2, p_2)) \tag{3.10}$$

Average KL-divergence for Normal distributions:

$$E_{\text{Mean-KL}}(\mathbf{g}, \mathbf{p}) = \frac{1}{n}\sum_{i=1}^{n} E_{\text{KL}}(\mathbf{p}(i), \mathbf{g}(i), \Sigma_{p(i)}, \Sigma_{g(i)}) \tag{3.11}$$

$$E_{\text{KL}}(\mathbf{g}, \mathbf{p}, \Sigma_p, \Sigma_g) = \frac{1}{2}(\Sigma_g^{-1}\Sigma_p + (\mathbf{p} - \mathbf{g})^T\Sigma_g^{-1}(\mathbf{p} - \mathbf{g}) - d - \ln(\Sigma_g^{-1}\Sigma_p)) \tag{3.12}$$

Above Σ_p is a diagonal matrix as we assume the predictor is uncorrelated, Σ_g is a per time step covariance derived from labels given for that timestep by multiple people; d is the number of dimensions considered.

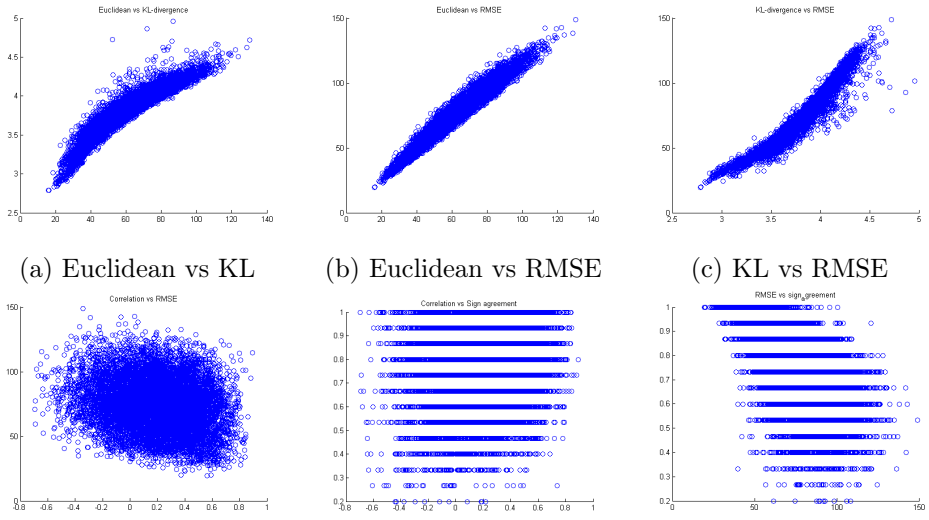
Finally we define a **combined version of SAGR**:

$$E_{\text{Sign agr}} = E_{\text{Sign agr}}(p_1, g_1) + E_{\text{Sign agr}}(p_2, g_2) \tag{3.13}$$

3.1.6. Correlation between different metrics

I was interested to see how well, if at all, the different evaluation metrics correlate between each other. There were two reasons for such evaluation. Firstly, I wanted to see if improving the performance as measured by one metric would inevitably lead to better performance as measured by other metrics. Secondly, knowing which metrics do not strongly correlate with each other would help in the selection process for my study—only metrics that do not strongly correlate with each other need to be tested for preference.

To achieve this, 100,000 sequences were generated—hypothetical predictions—that were based on the ground truth, and fell within 1 SD of the average label for each sample. I evaluated each sequence using the metrics described above and generated a set of scatter plots, shown in Figure 3.1, depicting the relationship between the different metrics. My findings were rather surprising. Correlation and SAGR metrics differ markedly from the other three and each other. Correlation is particularly distinct, and does not seem to be related to either the RMSE or SAGR metric



(a) Euclidean vs KL (b) Euclidean vs RMSE (c) KL vs RMSE

(d) Corr vs RMSE (e) Corr vs SAGR (f) RMSE vs SAGR

Figure 3.1: Scatter plots of relationships between metrics when comparing a noisy synthetic prediction with ground truth. Notice how Euclidean, KL-divergence and RMSE are related.

at all. Euclidean distance, RMSE and KL-divergence exhibited a stronger relationship, with a slightly larger divergence between KL-divergence and the other two. This is not particularly surprising as all 3 metrics are based on the Euclidean metric in one way or another. This allowed me to simplify my study a bit—I discarded Euclidean distance in favour of RMSE, as the two correlate strongly, and RMSE is a more popular metric in the field.

3.1.7. Defining a sequence

All of the above metrics except for the correlation coefficient are calculated on a per time-step basis, and are then averaged across the whole sequence. Correlation coefficient relies on the mean value of the sequence as well—in calculating $\overline{p}$ and $\overline{g}$)—so it becomes important how such a sequence is defined. In the Audio/Visual emotion recognition community the sequence is generally defined as a recording (or a part of a recording). A correlation score is then calculated for each of the recordings (*short correlation*). This is averaged across all of the sequences to provide a final evaluation metric. In the music community, however, it is more common to concatenate all of the individual songs into one long sequence and then compute the correlation for the whole set (*long correlation*).

At first glance, whether short correlation or long correlation is used does not seem to make much of a difference. Shortening the sequence for which correlation is computed and averaging those coefficients will inevitably lead to a lower correlation score. However, computing long correlation score might hide bad per-sequence predictions. For example, a predictor that is good at predicting the average position in valence space for a song can still get a high correlation score, even though it is bad at predicting change within a sequence (which is particularly interesting to me). In other words, high long correlation, for an emotion tracking

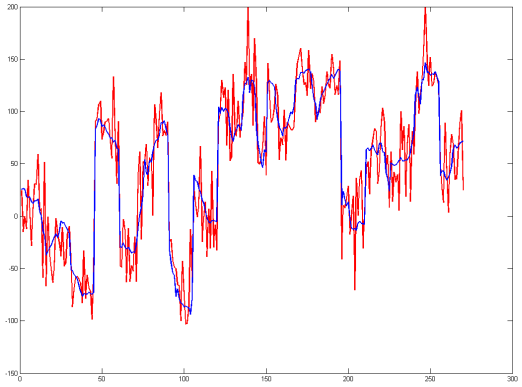


Figure 3.2: Example of a predictor with different correlation scores depending on how sequence is defined. Blue is the ground truth data, red is a hypothetical prediction. For this predictor if we take the overall sequence as one the correlation score $r = 0.93$, but if we take correlations of individual song extracts (18 sequences of 15 time-steps each) the average $r = 0.45$.

system, means capturing the overall emotion, while a high short correlation score means capturing the changes of emotion within a sequence, and the two do not have to be correlated. The effect of this is illustrated in Figure 3.2.

The different choice of the granularity used makes less of an impact for the other metrics, as their computation does not include any overall statistics, but the effect described is still present.

3.2. Implicitly preferred metric

As discussed in Section 3.1.6, the analysis of the most popular evaluation metrics led me to focus on the following four: correlation coefficient, RMSE, SAGR and KL-divergence. The next step was to design a study that would allow me to determine how people instinctively evaluate the goodness of an emotion trace.

3.2.1. Generating predictions

In order to evaluate how well a certain metric represents people's perception of emotion in music, I needed to be able to present the participants of my study with several different emotional traces, each of which is optimised for a particular metric. I chose to use a hypothetical predictor that always predicts the trace as centered around the ground truth but with added Gaussian noise (the standard deviation of the noise matching that of human labelers of the ground truth dataset). I believe this amount of noise is justified as we would expect a statistical approach to perform within the boundaries of human variation. Examples of such a noisy trace can be seen in Figures 3.3, 3.4, and 3.9. In order to not disadvantage the synthetic traces for sign agreement metric (as they can be extremely noisy but still produce the same sign agreement score) they were drawn as straight lines with transitions when sign changes.

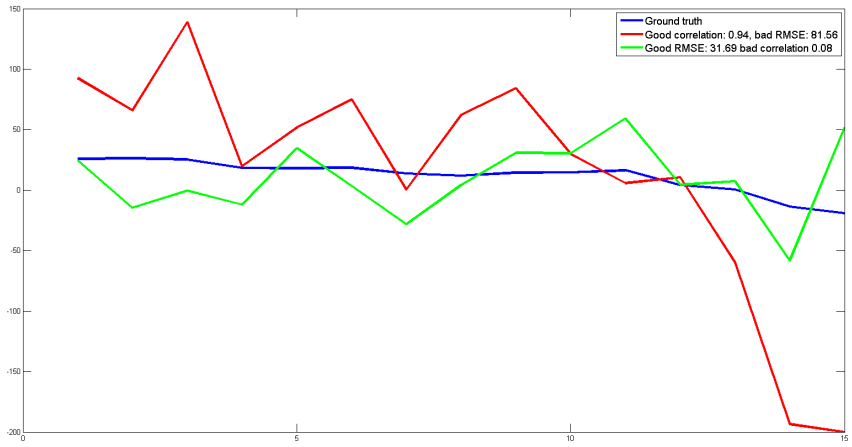


Figure 3.3: Sample synthetic traces. Blue is the ground truth, Red has a great correlation score (0.94), but bad RMSE (81.56), and green has a low RMSE (31.69), but bad correlation (0.08)

3.2.2. Optimising one metric over another

Once a sufficient number of noisy predictions is generated using the hypothetical predictor (I used 10^5 predictions) we can choose a prediction for a sequence that has the best score with a metric of interest when compared to the ground truth. So, for example, from the 10^5 generated noisy sequences I pick one that has the best correlation coefficient with the ground truth and use that for the further experiment. I do this for every metric I am interested in.

For the metrics I chose for the experiment (correlation, RMSE, SAGR, and KL-divergence) a sequence of predictions that optimised one metric never happened to be the one with the best score for another, hence just by generating noisy data I was able to pick predictions that have different scores for different metrics. For example: in Figure 3.3 both traces (red and green) have been generated by adding the same type of noise, however they resulted in two very different traces with very different metric scores—one has good correlation, but poor RMSE, and the other has good RMSE, but poor correlation.

3.2.3. Experimental design

There were several different questions that I wanted to answer with the study. First of all, I wanted to see whether people differentiate between or have preference for a particular way of optimizing (or evaluating) emotion traces. If so, I was interested to see if the preferred evaluation technique depended on a choice of a song. I was also interested to see if the preferred evaluation metric depended on the axis (arousal or valence) or the number of dimensions (one or two).

To achieve this goal, I designed the following study. Each participant was presented with 56 15-second extracts from a subset of songs used in MoodSwings dataset (see Section 2.4.1 for the description of the dataset). For each song I produced a

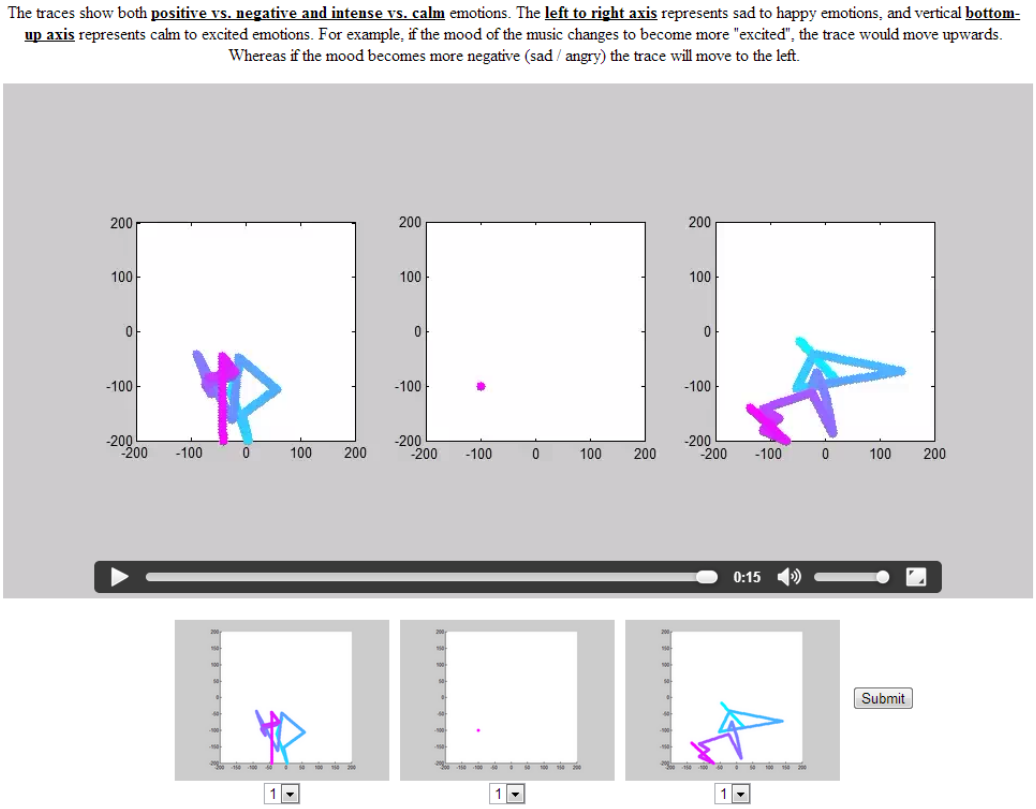


Figure 3.4: Screenshot of the study page. Instruction at the top, followed by a video and the static emotion traces.

video that displayed several emotion traces at the same time, synchronised with the audio extract (see Figure 3.4 for a screenshot of a study page). The participants were allowed to re-watch the video as many times as they wanted. Underneath the video, all the traces were presented in the static form (as they appear at the end of the video) with a drop-down selection for ordering them. The participants were forced to give a unique ordering for the traces, i.e. they were not allowed to say that any two or more of the traces were equally good.

Each trace for a song was based on a different evaluation metric—one optimized for correlation (best correlation, but higher RMSE, and lower SAGR), one for SAGR, RMSE and KL-divergence (see Section 3.2.2 for more details). The presentation order of the traces was randomized for each song, but fixed for all the participants.

The songs were split into three groups, and therefore each participant was presented with three different tasks. In the first part of the experiment, I had 18 songs with a focus on the arousal axis. The songs were chosen with as much change in the arousal values and little change in the valence values, based on the labels in MoodSwings dataset. The participants were shown and had to order the arousal traces only (see Figure 3.9 for an example of what such traces would look like). The second group of 12 songs had the exact opposite properties—some change in the valence and little chance in the arousal values. The participants were presented

with traces of affect on the valence axis only. The third task was focused on the change in emotion on both axes. The 26 traces used in the last part were shown in 2D and were colour-coded to represent time, changing from cool to warm colours (Figure 3.4).

The songs within each task were presented in a random order for each participant, but the order of the tasks remained the same. I hypothesized that one-dimensional emotion traces are easier to understand and deal with than two-dimensional ones and that arousal is easier to judge than valence. The tasks were ordered by difficulty (arousal, valence and then 2D) and this therefore allowed the participants to practice on an easier task before moving on to a more difficult one.

Pilot study

To evaluate the suitability of my experimental design, I first ran a pilot study. I recruited two participants from my research group, who were not aware of the design of the study or its purpose.

As explained in the Section 3.1.6, I first used 4 different evaluation metrics—correlation, RMSE, SAGR and KL-divergence. The design of the experiment followed the description above.

Both participants did the study individually. They were provided with a pair of headphones each and did the study in their own time. The instructions were given on the screen explaining the tasks. The experiment lasted approximately 30 mins for each participant.

The comments I received after the study confirmed that the task of evaluating 2D emotion traces was more difficult than 1D. The results also confirmed the appropriateness of my experimental design—there was a clear difference between the average rank for each of the evaluation metric.

Changes in the final study

After the success of my pilot study, I conducted the actual experiment with several changes—all based on the comments I received.

In the pilot study I found that even with only two participants, it was already clear that KL-divergence and RMSE achieve the same average rank—both per participant and overall. This, together with the theory described in Section 3.1.6 and the comments from the participants that it was often difficult to order 4 different traces, led me to decide to remove one of them. As RMSE is generally used for models dealing with one axis at a time, I kept RMSE as the third evaluation metric for the first two (one-dimensional) tasks. Similar reasoning led me to remove RMSE and keep KL-divergence for the third, two-dimensional, task.

I also made several changes to the instructions provided at the beginning of the study, making them more informative and clear. In addition, I provided the participants with a sheet explaining the meaning of arousal and valence axes. They were allowed to keep it and refer to it throughout the experiment. See Appendix A for the handout I provided in the study.

I had 20 participants (13 female and 7 male), recruited through a local ad-website and graduate-student newsletters. Each participant was required to come to my lab for the study and received a £10 Amazon voucher for their time. I had up to 5 participants doing the study at the same time, all in the same room, each provided with a pair of headphones and doing the study in their own time. All of the instructions were given on the screen, and apart from 2 participants, none of them required extra verbal explanations.

The participants were given an opportunity to leave comments after the study. Most of the feedback I received was positive—in general, participants enjoyed doing the study and found it interesting. Several participants said that they had some difficulty with the third task, and had to re-watch the videos several times.

3.2.4. Results

For the purpose of this study, I use the rankings for each song and each metric as numerical values ranging from one to three—1 being the most and 3 being the least preferred choice. This allows me to compute average rankings for each metric for each song, participant, or task. It also allows me to compare the means and check if any differences are statistically significant.

I split the analysis into two parts—two one-dimensional tasks, and one two-dimensional task. There are two reasons for this. Firstly, since I have used slightly different metrics for the two types of tasks, it was impossible to combine all of the data I had into one analysis. Secondly, I expected similar results/conclusions from the two one-dimensional tasks, while I expected the results to possibly differ between one-dimensional and two-dimensional tasks.

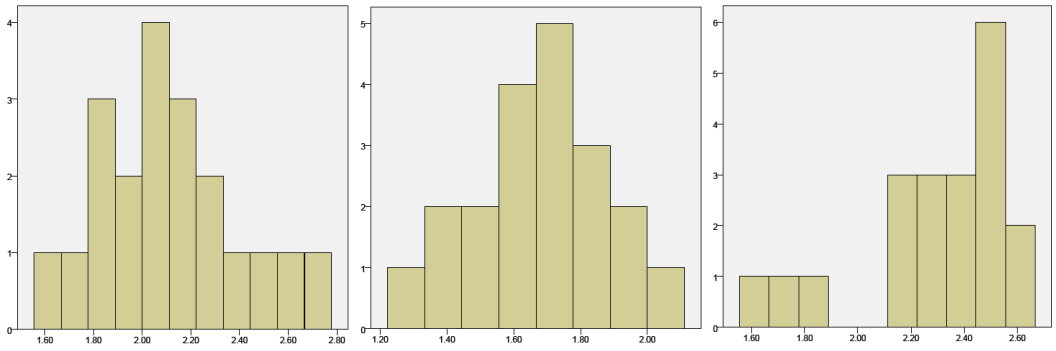
One-dimensional tasks

There are several questions I wanted to answer when looking at the data from the one-dimensional tasks. First of all, I wanted to check if there is any effect of the dimension on the average rank. Then within each dimension I wanted to check if the ranks are significantly different from each other, and if so, which one of them is preferred.

Normality

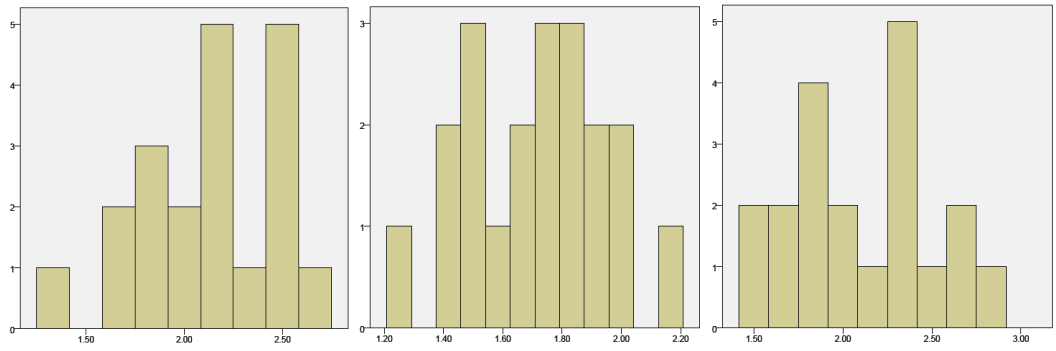
In order to answer these questions, I needed to check that my data is normally distributed, as many statistical tests require this. I calculated the average rank for each metric and each dimension per participant, i.e. we computed a 20x6 table (20 participants, 2 dimensions, 3 metrics) of mean ranks.

All but one (SAGR for arousal) of the distributions are approximately normally distributed (see Figures 3.5 and 3.6). This is confirmed by Kolmogorov-Smirnov test—there is a statistically significant difference between the SAGR data for arousal and the normal distribution ($D(20) = 0.21, p = 0.023$). On the other hand, there is no statistical difference between the normal distribution and any other datasets ($D(20) = 0.19, p > 0.05$ for RMSE for arousal and $D(20) = 0.12, p > 0.05$ for valence axes, $D(20) = 0.15, p > 0.05$ for correlation for arousal and $D(20) =$



(a) Correlation (b) RMSE (c) Sign-agreement

Figure 3.5: Arousal distributions for the three metrics



(a) Correlation (b) RMSE (c) Sign-agreement

Figure 3.6: Valence distributions for the three metrics

$0.12, p > 0.05$ for valence axes, and $D(20) = 0.14, p > 0.05$ for SAGR for valence axis).

When the data is aggregated over the two dimensions (giving 20x3 values), all three distributions show no statistically significant difference from the normal distribution.

ANOVA

A repeated measures within-subject factorial ANOVA with dimensions (2 levels) and metrics (3 levels) as factors show a small significant effect of dimension on the average rank ($F(1, 19) = 5.5, p = 0.030$). The effect of metrics, on the other hand, is much stronger ($F(2, 38) = 16.39, p < 0.001$), with no interaction between the two ($F(2, 38) = 1.785, p > 0.05$).

The pairwise comparison (with Bonferroni adjustment for multiple comparisons) reveals that there is a statistically significant difference between the average ranks for RMSE and correlation ($t(19) = -5.39, p < 0.001$), and RMSE and SAGR ($t(19) = -5.68, p < 0.001$), but no significant difference between correlation and SAGR ($t(19) = 0.75, p > 0.05$). The same conclusion can be observed in the box-and-whisker plot showing all 6 distributions (Figure 3.7). This is also confirmed by the fact that RMSE is selected as the top choice 43% of the time (SAGR—27%,

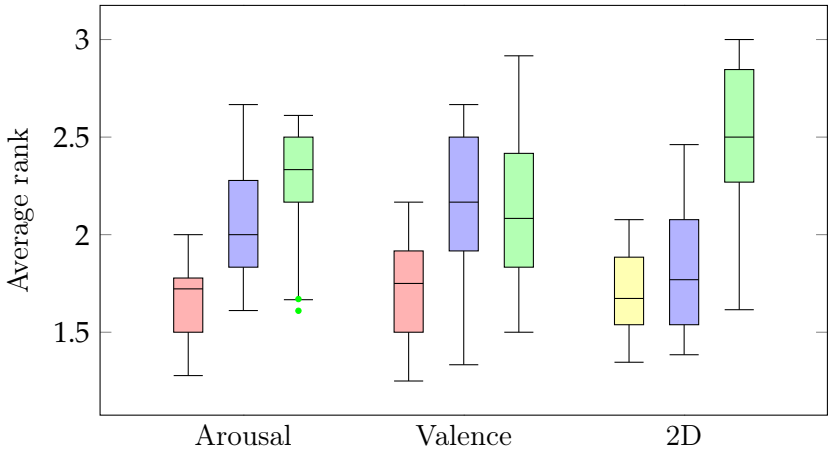


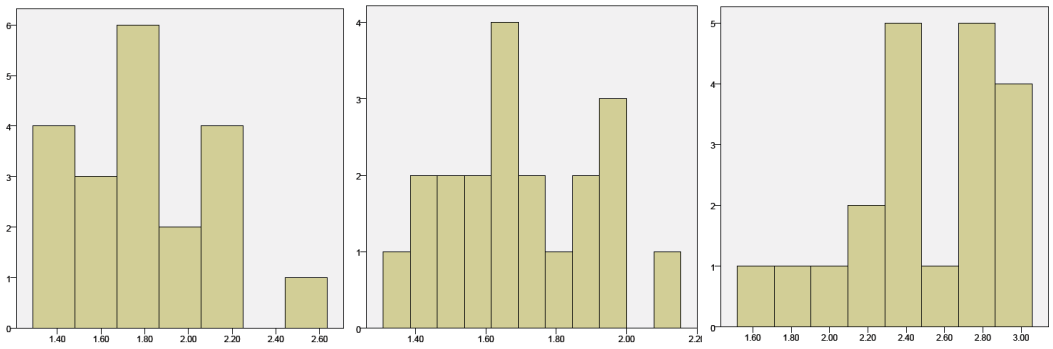
Figure 3.7: Average ranking of the three tasks. RMSE Correlation SAGR KL-divergence

correlation—30%).

Two-dimensional task

The questions I want to answer when looking at the two-dimensional task are the same as the ones from one-dimensional tasks. Mainly I am interested in seeing if there is a statistically significant difference between the average ranks of the different metrics. And if so, which is the preferred one.

Normality



(a) Correlation (b) KL-divergence (c) Sign-agreement

Figure 3.8: 2D-task distributions for the three metrics

Again, I first check if the data is normally distributed (Figure 3.8). I aggregate data in the same way as for the one-dimensional tasks—average the rank for each metric for each participant. This time all three datasets are normally distributed—the Kolmogorov-Smirinov test showed no statistically significant difference between the three sets and the Normal distribution ($D(20) = 0.09, p > 0.05$ for correlation, $D(20) = 0.13, p > 0.05$ for SAGR and $D(20) = 0.12, p > 0.05$ for KL-divergence).

ANOVA

A one-way repeated-measures ANOVA with metrics (3 levels) as factors shows that there is a strong, statistically significant effect of metrics on the average rank ($F(2, 38) = 28.55, p < 0.001$). Pairwise comparisons between the three metrics (with Bonferroni adjustment for multiple comparisons) reveal that the average rank for SAGR is statistically significantly different from correlation ($t(19) = 4.89, p < 0.001$) and KL-divergence ($t(19) = 6.60, p < 0.001$). However, there is no statistically significant difference between the average ranks of KL-divergence and correlation ($t(19) = 1.29, p > 0.05$). This can also be observed by a visual inspection of the box-and-whisker plot (Figure 3.7). KL-divergence and correlation are also the most frequent top choices—38% and 45% (respectively) of the time, with SAGR being chosen as the most preferred option 17% of the time.

Further analysis

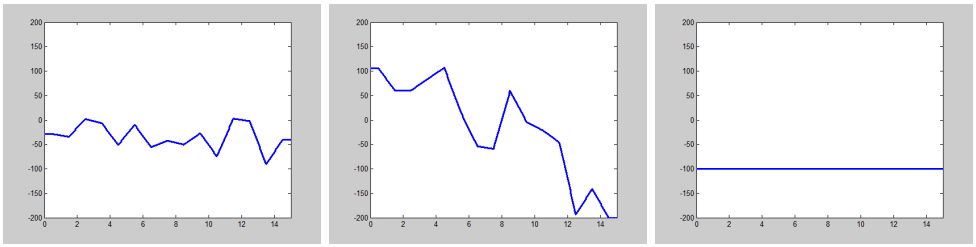
As explained in Section 3.1.7, there is a notable difference between short and long correlation. As a post-hoc analysis, I generated several long sequences, by taking a trace for each song and concatenating them all into once sequence. I then looked at the long correlation score of that trace with the one composed of ground truth labels. For both one-dimensional tasks I generated 4 traces: one composed of the traces from people's top choices for each song, as well as 3 sequences composed of traces for each evaluation metric. For arousal, the long correlation of the top choice reached 0.87, while RMSE-optimized traces had correlation of 0.93. The lowest one was from short correlation optimized traces (0.77), with even SAGR scoring higher (0.82). Similar results are seen for valence (top choice – 0.80, short corr. – 0.72, RMSE – 0.89 and SAGR – 0.89). This analysis clearly shows the difference between short and long correlation, and that the two do not have to be correlated—for both tasks, short correlation-optimised traces achieved the lowest long correlation score.

I also wanted to consider whether or not the preferred choice of evaluation metric might depend on a song in question. To investigate this question, I took the average rank for each metric over each song, rather than over the participants. I then inspected the results to see if there are any exceptions.

Even though the majority of songs seem to follow the trends described in Sections 3.2.4 and 3.2.4, there are some examples of songs with a different preference for evaluation metric. Occasionally, participants were choosing SAGR over the other two metrics. As can be seen from an example in Figure 3.9, which depicts one of such songs, these songs tend show less variation in the expressed emotion and SAGR tends to show a flat line.

3.2.5. Discussion

The discussion of the results of the study described in this chapter can be split into two parts. The main results allow me to suggest which metrics should be used when evaluating and developing music emotion prediction algorithms. The analysis of literature and more minor observations from the study also encourage me to suggest some further guidelines for future work.



(a) RMSE (b) Correlation (c) Sign-agreement

Figure 3.9: Example valence trace of a song used in the experiment

Choice of evaluation metrics

My study indicates that RMSE corresponds to people's perception of 1D emotion in music the best. It should therefore be used for optimizing algorithms to estimate one-dimensional models (Section 3.2.4), and that it is also the most appropriate metric for reporting results.

For two-dimensional models the situation is less straightforward. The analysis of the results from the third task (Section 3.2.4) indicate that both correlation and KL-divergence were equally preferred by the participants. As a choice between the two still needs to be made, I would suggest using KL-divergence for the development of algorithms, as it is more similar to the preferred choice for one-dimensional models. I would then advise researchers to use both metrics when reporting the results.

Other considerations

There are several other issues that might be worth considering when approaching the problem of emotion prediction.

First of all, the fact that RMSE was the preferred choice as an optimization metric identifies two things participants cared about. It seems that when judging the emotional content of a song, participants expect to see not only the relative change of emotion within a song (as correlation would suggest), but also the absolute position of the trace in the arousal-valence space. This has implications not only on the choice of evaluation metrics to use, but also on the kind of models that should be investigated in future work.

Another observation is that there was a (small) number of songs where sign-agreement was preferred over the other metrics (Section 3.2.4). It only seems to occur when there is little change in the expressed emotion of a song—in which case sign-agreement displays a flat line, while other metrics fluctuate around it. This suggests that a level of smoothing might be preferable when predicting emotion or as a post-processing step when displaying the results.

I also urge against using only long (and squared) correlation as an evaluation metric, as it hides important information about the performance of an algorithm (Section 3.1.7). It also does not seem to relate well to people's preferences (Section 3.2.4)—the long correlation of neither the people's preferred traces nor the (short)

correlation-optimised traces was the highest that could have been achieved. It is not clear if long correlation has any use at all, so I would strongly suggest at least to report short correlation in addition to long correlation.

As it is possible to achieve good results in one metric while bad results in other metrics, I advise reporting the results using several metrics. This would give a better understanding of the general behaviour of an algorithm. In addition to that, I urge researchers to give the formulas of the metrics used in the evaluation. It is often not clear which exact evaluation metrics are used to describe the results (short versus long correlation, normalised versus non-normalised, etc.), making it more difficult to compare different studies.

3.3. Conclusions

In this chapter I gave a summary of the different evaluation metrics that are used in the field of affective computing in general and emotion recognition in music in particular. I identified the problems with the way the evaluation has been done so far and identified a set of evaluation metrics that are the most popular. I designed and executed a novel study intended to identify people's instinctive preference for a particular evaluation metric when applied to continuous dimensional musical emotion representation.

The results of the study strongly suggest that RMSE is the most appropriate evaluation metric when used for a one-dimensional task. For a two-dimensional task KL-divergence (related to RMSE) and correlation are most appropriate. This study is the first study in the field that tried to justify the choice of the evaluation metric based on experiments and is able not only to identify the most appropriate metric, but also to make some suggestions with regards to algorithm design, that are based on the study findings.

The conclusions I have reached and suggestions I have made can obviously only be directly applied to the field of emotion prediction in music. Similar studies could and should be used to check if the same trends occur in other fields of affective computing, as well as for different types of representation and different types of noise added to the ground truth. I expect that similar conclusions will be drawn, but a more comprehensive comparison across different fields and using different options will provide results that are interesting either way.

Feature vector engineering

4

For every problem that lends itself well to a machine learning approach, there are always three parts that can be improved to reach a better solution: data, features and the algorithm. In this chapter, I will describe the work I have done on improving the features used for emotion recognition in music. I will describe the features that I have used and extracted and the different representations I have used to extract more information from the raw data. A lot of these feature representations are novel and as far as I am aware have never been used in the field of emotion recognition in music.

Some of the work described in this chapter has been published in:

Emotion tracking in music using continuous conditional random fields and baseline feature representation, Vaiva Imbrasaitė, Tadas Baltrušaitis, Peter Robinson, *AAM workshop, IEEE International Conference on Multimedia, San Jose, CA, July 2013*

Absolute or relative? A new approach to building feature vectors for emotion tracking in music, Vaiva Imbrasaitė, Peter Robinson, *International Conference on Music & Emotion, Jyväskylä, Finland, June 2013*

4.1. Features used

There were three sources of features that I used for building my feature vectors.

The first one came packaged together with the ground truth in the MoodSwings dataset [Speck et al., 2011]: it contained a set of low-level features (MFCC, chroma, spectral contrast and spectral descriptors) extracted with published scripts and also a set of high level features (timbre, pitch, loudness) extracted with EchoNest[1]. I chose not to use the EchoNest features in any of my experiments, since they have been extracted with proprietary software that does not provide clear documentation or explanation of how the features are extracted. Using such features would make the work less reproducible and results less robust.

[1] https://developer.echonest.com/

The second set of features was extracted by the OpenSMILE[2] tool with a modified ComParE script which was originally written by Steidl et al. [2013]. The original script produces over 6000 features and it was used for the MediaEval 2013 Emotion in Music task by Weninger and Eyben [2013] (see Sections 2.3 and 2.4.2) with great results. As I was trying to maintain the same experimental conditions to test all the models, and I knew that CCNF (see Section 5.3) suffers and fails to converge when dealing with that many features, I modified the script to reduce the number of various statistical descriptors, but left the same main type of features reducing the total number of features to 150.

OpenSMILE (Speech & Music Interpretation by Large-space Extraction) is a C++ library for extracting features deemed to be useful for Speech Processing and Music Information Retrieval systems. The features are generally quite low-level (including MFCC, loudness, energy, mel-spectra, chroma, etc.), but it also provides a large array of statistical functionals for higher-order processing of features. It is cross-platform, works well for large-scale processing, and can export the features in several widely used (machine learning) datafile formats.

Finally, another tool worth mentioning is the MIRtoolbox[3]. MIRtoolbox is a Matlab library of feature extraction functions that are specifically designed to be used by researchers in the field of Music Information Retrieval. The toolbox covers a wide range of functions starting from low-level ones (like spectrogram), to high-level ones (like harmonicity). It also decomposes some of the more complicated, higher-level features and exposes the different stages required in the extraction of the feature values. In addition to that, MIRtoolbox provides a package of statistical functions as well as visualisation techniques for most features. This toolbox was used to extract the features for the baseline method in the MediaEval 2014 Emotion in Music task (see Sections 2.3 and 2.4.3), which have also been used in this work. The feature set consists of 5 features: spectral flux, harmonic change detection function, loudness, roughness and zero crossing rate.

Below are the descriptions of some of the more common features that were used at one point or another in this work.

4.1.1. Chroma

A chromagram is a musically inspired set of features. The features represent a many-to-one mapping between the spectrum of a signal—a song or an extract from a song in this case—(example of which can be seen in Figure 4.1) and the 12 distinct semitones (or chroma) that constitute an octave (depicted in Figure 4.2). A chromagram is very similar to a spectrogram as it provides a sequence of short-time chroma frames that represent the whole signal in a modified frequency domain. Chromagram has a much higher information density (notice the low presence of higher frequencies in Figure 4.1 with most of fluctuations appearing in the bottom quarter of the spectrogram) than a spectrogram with an additional advantage of that information being musically relevant. In addition, as the chromagram

[2] http://opensmile.sourceforge.net/

[3] https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox

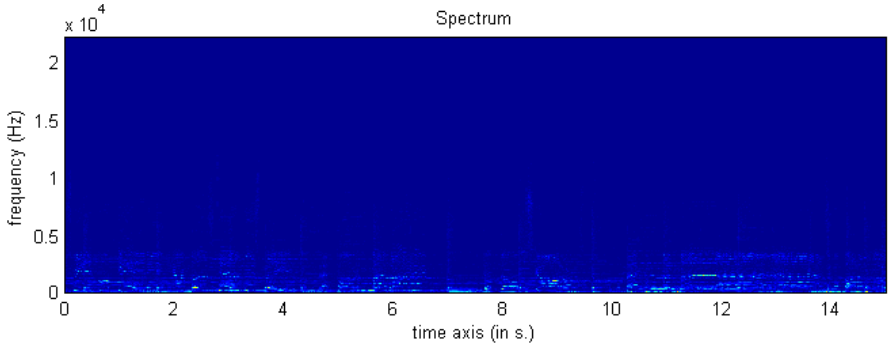


Figure 4.1: An image of a spectrogram of a song

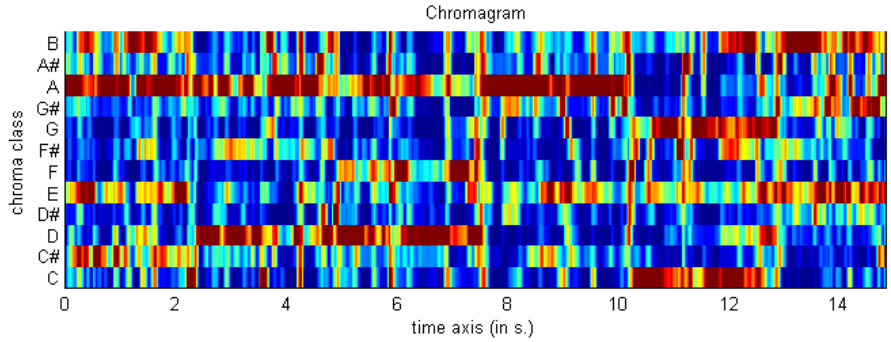


Figure 4.2: An image of a chromagram of a song

gives a probability distribution of various notes, it can be used to construct a probability distribution of different chords as well as the musical key.

As the same note in two different octaves is perceived as musically similar, chromagram gives a good summary of the musical content present in a signal. While the use of such features for the emotion recognition in music intuitively makes sense, it has also been tested in an experiment. Schmidt et al. [2012] extracted chromagrams from a set of Beatles songs and then re-synthesized an audio from them. Despite the loss of information that happens when converting a spectrogram into a chromagram, the opposite process still results in something that resembles music. Pairs of these resynthesized songs were then given to people to listen and they were asked to compare their emotional content using the dimensional representation—they were asked to identify which of the two songs was more positive (valence) and which was more intense (arousal). The same was done with the original songs and the results in the two conditions were compared. Schmidt et al. [2012] found a positive correlation between the two conditions indicating that chromagram in fact encodes some emotional content of the song—the normalised difference error between the two conditions was 0.120 for valence and 0.121 for arousal, showing that chroma is similarly important to both axes.

4.1.2. MFCC

Mel Frequency Cepstrum (MFC) is another power-spectrum based representation of an audio signal. It is based on a log power spectrum on a Mel scale of frequency.

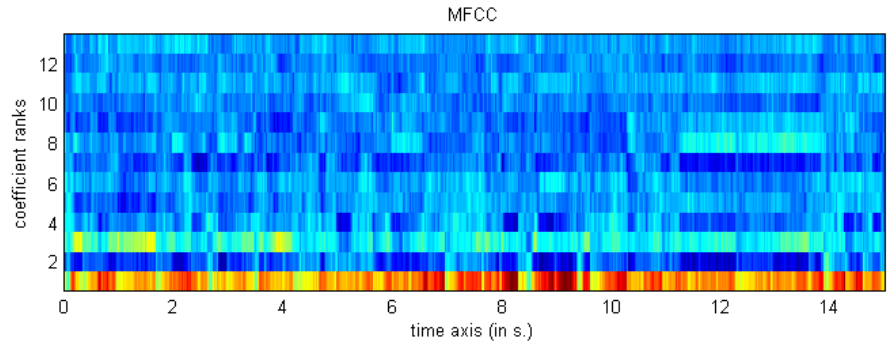


Figure 4.3: An image of the MFCC of a song

This representation is biologically inspired and is meant to mimic the way human ear perceives sound. Mel Frequency Cepstral Coefficients (MFCC), that make up the MFC, is a state-of-the-art feature set that was initially used for speech recognition, but has now been adopted and widely used in the music information retrieval community.

MFCC extraction process consists of 5 steps:

1. Transform the signal into a sequence of short term frames
2. Take the Fourier transform of the resulting signal
3. Map the power spectra onto the mel scale using a filterbank of triangular overlapping windows
4. Take the log of the mel frequencies
5. Take the discrete cosine transform (DCT) of the mel log energies and keep the 2-13 coefficients discarding the rest

Just like in the other, similar forms of signal representation (such as spectrogram or chromagram), the length of the frame is very short—20-40ms. This way we can consider the signal reasonably stable while still having enough samples for a reliable calculations. The Fourier transform is supposed to mimic the workings of the human cochlea which responds to a noise by vibrating at different locations that correspond to different frequency ranges. As the human ear loses its discriminative powers as the frequency gets higher, it is important to map the frequencies to a different scale. The Mel scale consists of a set of frequency bands that are very narrow around 0 Hz and get increasingly wide as the frequency grows higher. It can be calculated by the formula $M(f) = 1125 \ln(1 + f/700)$. The log step is inspired by the fact that the human ear responds to loudness on an exponential, and not a linear scale—to double the perceived loudness, the the energy in the signal has to be increased 8 times. This is approximated by taking the logarithm (instead of a cube root) to allow standard channel normalisation techniques. Finally, DCT is performed in order to minimise the correlation between the different filterbank energies (which are overlapping and would therefore correlate), so that it could successfully be used for various machine learning methods, such as Hidden Markov Models. The first coefficient is discarded as it represents the general

loudness of the signal, and the higher coefficients are discarded because they represent the faster changes in the signal and have been shown to be of little use. The middle 20 (usually) coefficients are then used to represent the signal. An example of the MFCC representation of a song can be seen in Figure 4.3 (compared with the same song's spectrogram in Figure 4.1 and chromagram in Figure 4.2).

Unlike in the chromagram's case, it is not immediately obvious that MFCCs can directly encode emotional content of a song. To test this, Schmidt et al. [2012] included the comparison between the signal recreated from MFCCs and the original songs in their study (see section 4.1.1 for a more detailed explanation). Similarly to the chromagram, the signal recreated from MFCCs had a positive correlation with the emotional ratings of the songs, but in this case the effect was stronger for arousal, than for valence—the normalised difference error between the ratings for the originals and the resynthesised songs was 0.133 for valence and only 0.104 for arousal. The resynthesised songs lose their melodic line, but maintain a strong rhythm, which explains the stronger effect on the perceived arousal.

4.1.3. Spectral Contrast

Octave-Based Spectral Contrast (OBSC) is another popular feature used in MIR. It was first introduced by Jiang et al. [2002] to the task of genre recognition. OBSC is design to capture spectral shape characteristics, to describe the song's spectral distribution. The idea is that spectral peaks are supposed to represent the harmonic components, while the spectral valleys correspond to non-harmonic components and noise, so the difference between the two should reflect the relative distribution of the harmonic and non-harmonic components.

The raw Spectral Contrast features are extracted by first performing a Fourier transform of the signal, and then mapping the resulting frequencies into octave-based bands. The values are then sorted in descending order and the sum of the largest 2% of the values is considered the peak and the sum of the lowest 2% is considered the valley of that band. In the original paper, the peak sum value is then replaced by the difference between the peak and the valley (the spectral contrast) and the valley sum value is kept. The log of both is then taken, and Karhunen-Loeve (KL) transform is applied to the entire feature vector. This makes the whole process similar to the process of extracting MFCCs—instead of mapping frequencies onto a mel-scale, an octave-based scale is used, and the DCT is replaced by KL transform. The main difference is the processing of the spectral values which in MFCC extraction are simply summed in each band, while in Spectral Contrast only the peak and valley information is kept. Jiang et al. [2002] showed that a machine learning model using Spectral Contrast as its feature vector outperforms one that uses MFCCs or MFCCs with Energy terms in the task of genre recognition for a particular dataset.

There are variations of the Spectral Contrast feature. The Spectral Contrast feature used in the MoodSwings dataset, for example, consists of 14 features: it contains the peak and the valley values for 7 octave-based bands, that are extracted in the same way as described above.

4.1.4. Statistical Spectrum Descriptors

Statistical Spectrum Descriptors (SSD) is a set of statistical measures applied to the frequency spectrum of a signal. In addition to other purposes, it can serve as a good approximation of the rhythmical features of a song, which are generally a good predictor of the arousal values. The actual measures used are not consistent between the papers that claim to be using SSD as part of the feature vector, so I will describe some of the more common ones.

Spectral centroid describes the shape of a spectrogram, by giving the mean or the geometric centre of the spectral distribution. While spectral centroid shows the tendency of a distribution, *spectral rolloff* essentially describes the spread of a spectrogram. As the shape of such distributions tends to be heavily skewed towards the high frequencies, spectral rolloff identifies the frequency under which most of the energy is contained (usually either 85% or 95%). Another way of describing the shape of a distribution is through the *spectral flatness* feature. Spectral flatness tells us whether the distribution is smooth or spiky through the ratio of the geometric mean to the arithmetic mean.

Another important factor describing a spectrogram is its temporal features. *Spectral flux* identifies the changes within the spectrogram over time. It is computed by calculating the distances of the spectrum between each successive frames. This way any sudden changes within the spectrogram will be immediately obvious, while a slow build-up can be identified through the second order statistics.

Entropy provides a general description of a curve. When applied to a spectrogram, it tells us whether the spectrum contains any predominant peaks—maximal entropy corresponds to extremely flat curves, and minimal entropy is observed when there is a single sharp peak in the signal. Care must be taken to make the spectral entropy calculation independent of the signal length, but as most of these descriptors are calculated from windowed signal with fixed window length it is not generally a problem.

Noisiness in an audio signal can be approximated by its *zero crossing rate*. By definition, zero crossing rate tells us the rate at which a signal crosses the X-axis. When used on a waveform of an audio signal, it will count the number of sign changes and will give a good indication of noisiness. Zero crossing rate is a good indicator of the presence of speech.

The MoodSwings dataset contains only 4 statistical descriptors in its SSD feature set: spectral centroid, spectral flux, spectral rolloff, and spectral flatness, while the datasets based on OpenSmile extracted features also contain spectral entropy, variance, skewness, kurtosis and slope. MIRtoolbox can provide a similar set of statistical signal descriptors including: zero-crossings rate, spectral centroid, spread, skewness, kurtosis, flatness and entropy.

4.1.5. Other features

Harmonic Change Detection Function (HCDF) is a method for the computation of tonal centroid flux introduced by Harte et al. [2006]. The changes in the harmonic

content of a song are detected through a 4-step process. First, a Q-Transform is used to convert the audio signal into the frequency domain. The Q-Transform is similar to the Fourier Transform, but produces a mapping that is closer to that of a human auditory system—the resolution for higher frequencies is lower, but fewer bins are used to cover the range of frequencies of interest (similarly to human hearing). The result is a 36-bins-per-octave transform covering 5 octaves. The next step is to map the spectrogram onto a 12-bin chromagram (see Section 4.1.1 above). The third step is the tonal centroid calculation, which transforms the 12-dimensional chromagram into a 6-dimensional tonal centroid vector. It is achieved by a multiplication of the chroma vector and a transformation matrix, that is then normalised to prevent numerical instability. The coefficients in the transformation matrix are taken from three circles representing the relationships between different pitch classes (fifths, minor thirds and major thirds). The tonal centroid vector is then smoothed over time, and the HCDF is defined as the overall rate of change of the smoothed tonal centroid signal. Euclidean distance between the vector of the previous and the subsequent frame is calculated—a peak in the resulting signal would therefore indicate a change between two harmonically stable musical regions. Based on musicology research into the effects that various musical features have on perceived emotion (Table 2.1) we know that changes in musical features can have a strong and varied effect, which justifies the use of this function for the task of MER.

Loudness is also sometimes used as a feature in MIR. It is measured as the sound pressure of an audio signal and can be expressed as the logarithmic sound pressure level, measured in decibels. The most common description of loudness is Zwicker's loudness—to calculate, the spectrogram is first mapped onto the psychoacoustic Bark scale, then the masking effects of low frequencies are taken into account; finally, the square root of the sound pressure is taken to achieve the final loudness level. Loudness can be a useful feature for within-song loudness variations, but the overall loudness level should be normalised across all the songs in the dataset to minimise the variations in recordings that might lead to an external effect on the perceived emotion.

Sharpness is another psychoacoustic feature providing a tonal description—it can be seen as the "tone colour" with higher frequencies being considered as sharper and therefore affecting the perceived arousal of a song. The sharpness of a signal can easily be calculated from the loudness pattern as the first momentum of the signal that is manipulated to emphasize the higher frequencies.

Another useful feature that can easily be extracted from an audio file is *roughness*, or sensory dissonance. This phenomenon occurs when two sinusoids are close in frequency and can be estimated by the ratio between each pair of sinusoids. The total roughness of an audio can therefore be computed by first extracting the spectrogram, identifying its peaks and then taking the average of the dissonance between all possible pairs of peaks.

A lot of other, higher level features, such as rhythm, are generally poorly defined and/or are difficult to extract and do not yet have established algorithms for extraction. While most of the time they are not used explicitly as features, they are

undoubtedly implicitly encoded in other features extracted from music.

4.2. Methodology and the baseline method

For each source of features, I used the same principle for constructing the baseline method. The standard approach used in the fields of both continuous and static emotion recognition is to use the audio features only, combined with support vector regression (SVR) or support vector machines (SVM) for classification problems. I used the LIBSVM [Chang and Lin, 2001] implementation of support vector regression for all of my experiments.

The feature vector that I used for the baseline method is based on the bag-of-frames approach. It consists of the audio features averaged over a 1s window—the mean for each feature (plus the SD for the original MoodSwings featureset). There is only one feature vector for each second of the song (so 15 training/testing samples for each extract that lasts 15 s), labeled with the average valence or average arousal value. The whole featureset is normalised so that the maximum value for each feature over the whole featureset is 1 and the minimum is -1. Normalisation is important for machine learning techniques when the range of values differs a lot—if we take MFCC coefficients, for example, the values of the first coefficients can differ from the values of the last coefficients by several orders of magnitude. In such a case, any changes in the higher coefficients would be hidden by the changes in the lower coefficients, and a machine learning algorithm would fail to extract meaningful patterns.

When using SVR, there is a set of kernels to choose from (see Section 5.1.1 the description of SVR and the kernels, as well as the comparison of the performance of the various kernels). Unless stated otherwise, all results are reported using SVR with radial-basis function (RBF) kernel, as this kernel consistently performed better than the other kernels. Two support vector regressors are trained—one for the arousal and one for the valence axes.

All the reported results, unless stated otherwise, are based on 5-fold cross validation. The whole dataset is split into 5 parts—4 of them are used for training, and then the model is tested on the 5th part. The the process is repeated 5 times, each time selecting a different part for testing on. The results are then averaged over the 5 folds. This way the model is tested on all the data available without ever being trained on the same data it is tested on. To improve the stability and reliability of the results, this process was sometimes repeated several times. Whenever several methods are compared, the distribution of songs over the folds is kept the same across all the experiments cited, to make the results more comparable.

The last issue to consider is the choice of various coefficients needed for training a SVR model. One of the most common approaches for picking the values for all the coefficients is to use grid-search—all possible combinations of coefficient values within a set region are tried and the one that produces the best results is picked. In order to minimise the risk of over-fitting, I used cross-validation for that too. That is, for each training set I use grid-search combined with cross-validation within that fold to pick the best values of the coefficients, and then I use those

Table 4.1: Artist-, album- and song-effect in the original MoodSwings dataset.

	No constraints		Song-level split		Album-level split		Artist-level split	
	Corr.	MSE	Corr.	MSE	Corr.	MSE	Corr.	MSE
Arousal	0.69	0.033	0.64	0.039	0.65	0.038	0.64	0.038
Valence	0.34	0.038	0.25	0.045	0.26	0.045	0.23	0.046

coefficients for training the whole training set. This way, the training set (whether for grid-search or for final results) is never used for evaluation. As the RBF kernel has two hyper-parameters that need to be picked for training, a 2-dimensional grid-search is used with the C parameter ranging between 2^{-7} and 2^{3} at every power of 2, and g ranging between 2^{-9} and 2^{-1} at every power of 2.

4.2.1. Album effect

I experimented with three different ways of distributing the songs between the folds (the effect on the squared correlation of the baseline method is shown in Table 4.1). As each song is split into individual time windows with their respective feature vectors, the most obvious requirement is to keep all the feature vectors from a song in the same fold, to ensure that the model is not overfitting to individual songs. For the baseline method, this lowers the squared correlation coefficient (R^2) from 0.34 to 0.25 for valence and 0.69 to 0.64 for arousal, and increases the mean squared error (MSE) from 0.038 to 0.045 for valence and from 0.032 to 0.039 for arousal.

Another factor worth considering is making sure that songs from the same album are all within a single fold. It has been reported and it is now widely accepted that the so called "album effect", first described by Kim et al. [2006], can artificially improve the performance as machine learning models overfit to a particular set of post production techniques used on an album. Removing the album effect did not make any difference to the results of the baseline method with the original MoodSwings dataset. This is probably due to the fact that a large majority of songs come from unique albums—the 240 songs present in the dataset come from 200 different albums, and so the overlap and therefore the room for overfitting is not large.

The third approach I tested was to make sure that all the songs from the same artist are within the same fold. Unsurprisingly, there is often statistically significant correlation between artists and mood in music [Hu and Downie, 2007], which, I expected, might lead to some overfitting. Again, it did not have a noticeable effect on the results with the baseline method, which is most likely because the dataset is fairly well balanced even for the artists—the 240 songs used were recorded by 156 different artists. It could also be argued that this restriction is unnecessarily strict—in real life, a fully trained system is unlikely to receive unseen songs from an album that it was trained on, but is definitely expected to analyze unseen songs from an artist that it has seen before. For these reasons, the song-level cross-validation was used for all of my experiments.

4.2.2. Evaluation metrics

Based on the findings described in Chapter 3, all of the results here and elsewhere throughout the dissertation are reported using both the short (computed over each song individually) and the long (computer by concatenating all songs into one) version of both correlation and RMSE (see Section 3.1.5 for exact mathematical formulas for calculating these). All the metrics are computed for each fold individually and then averaged over all folds. Long metrics are generally computed only for the purpose of comparison with other work, and short metrics are reported because I believe them to be more representative and indicative of the performance of the algorithms. Moreover, short correlation is kept un-squared so as not to hide inconsistent performance of an algorithm.

Note that lower RMSE values are considered better, and the opposite is true for correlation.

4.2.3. Feature sets

The combination of available datasets and possible sets of features that can be extracted from them makes it unfeasible to test all combinations with all the different methods suggested in this thesis. The main dataset used throughout the dissertation is MoodSwings (see Sections 4.1 and 2.4.1 for a description). As it comes with a pre-extracted set of features, it was an obvious choice for the initial experiments. Unfortunately, some of the experiments described in further chapters require manipulation of the actual audio, which means that the features need to be re-extracted. For that purpose I used the OpenSMILE script (described in Section 4.1). Table 4.2 shows the results achieved with the baseline model using SVR with the RBF kernel and the basic feature representation using the two featuresets. As we can see from the results, the original MoodSwings featureset achieves better results, especially for the long correlation metric, but the difference is much smaller for both the short and the long RMSE. Given that the improvement of the different methods proposed in this and further chapters have a much bigger impact than the reported difference between the two featuresets, and for the sake of consistency between the different experiments throughout the thesis, the OpenSMILE featureset extracted from the MoodSwings dataset is used, unless stated otherwise.

Table 4.2: Comparison of the standard featureset extracted from MoodSwings dataset and one extracted using OpenSMILE script using the baseline SVR model

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
Standard	0.194	0.178	0.645	0.011	0.220	0.186	0.211	0.007
OpenSMILE	0.206	0.188	0.610	0.014	0.217	0.189	0.224	0.048

4.3. Time delay/future window

The main drawback of the baseline model stems from the bag-of-frames approach. As each frame is considered independently, all of the temporal information gets

lost and the relationship between different frames is completely ignored. In addition to that, Schubert [2004] has showed that a change in some musical features takes longer to affect our perception of emotion than for other features, and so only looking at 1 s worth of acoustic features would make it impossible to capture such changes.

In order to address these issues I made the first modification to the feature vector of the baseline approach—I included the audio features from several one-second feature vectors. I experimented with two directions of such inclusion: a delay window (when the features of previous frames are included) to encode the changes that already happened, and a future window (when features of upcoming frames are included) to encode anticipation and the knowledge of the future, especially when the song is known already. In addition to that, I varied the sizes of these windows, both symmetrical and asymmetrical—from 1 s to 5 s for both the valence and the arousal axes.

As expected, including temporal information in the feature vector improved the performance of both the valence and the arousal models, although the effect of the two directions and on the two axes is quite different. First of all, Table 4.3 clearly shows that the effect of adding future samples into the feature vector has little (if any) effect on valence results and might even make them worse, and it does not do much more for arousal either. The addition of future samples is actually quite detrimental for the short correlation results, and we can see mixed results in other metrics when increasing the size of the future window that is included in the vector. So overall, future window does not seem to be much of an improvement over the basic representation.

The effect of using a delay window when building the feature vector is much more consistent and it follows a much clearer trend. We can see the improvement of increasing the window for both axes and it seems to plateau at around 4-5 s. The effect on the arousal axis is much stronger than that on the valence axis—both short and long RMSE for the arousal axis are improved by 0.2 and the long correlation is increased by over 10%, while the effect on the short correlation is immense and, as we will see throughout the rest of the dissertation, difficult to match with any other technique. The effect on the valence axis is less impressive, but still present: there is a small improvement on long and slightly larger improvement on short RMSE, but the effect on correlation is similarly large—if anything, it is actually larger, as the long correlation is improved by over 20%, while the short correlation nearly catches up with the short correlation for arousal.

Finally, the symmetric combination of the future and the delay window brings some improvement, but the results are a bit more mixed than for the delay window only. We can see a similar trend for arousal with similar results that plateaus at 4-5 s. The effect on the valence axis is less reassuring and seems to be more of a combination of the effects of the future and the delay windows—2 s window seems to improve over the basic representation results, but there seems to be little consistency of the effects on either side of it.

Table 4.3: Results for the time delay/future window feature representation, standard and short metrics

	Ar.				Val.			
	RMS	RMSs	Corr	Corrs	RMS	RMSs	Corr	Corrs
Basic	0.206	0.188	0.610	0.014	0.217	0.189	0.224	0.048
Delay 1s	0.200	0.181	0.634	0.095	0.214	0.185	0.241	0.094
Delay 2s	0.193	0.173	0.659	0.148	0.213	0.183	0.247	0.129
Delay 3s	0.190	0.170	0.671	0.177	0.214	0.185	0.253	0.140
Delay 4s	0.188	0.168	0.675	**0.206**	0.213	0.183	**0.274**	0.176
Delay 5s	**0.187**	**0.166**	**0.679**	0.198	0.214	**0.182**	0.271	**0.193**
Future 1s	0.203	0.184	0.623	-0.004	**0.215**	**0.186**	**0.235**	**0.026**
Future 2s	0.200	0.179	0.633	-0.023	0.216	0.187	**0.235**	-0.001
Future 3s	0.199	0.178	0.636	-0.020	0.217	0.188	0.229	-0.007
Future 4s	0.199	**0.177**	**0.639**	-0.040	0.218	0.190	0.234	-0.021
Future 5s	0.199	**0.177**	**0.639**	-0.029	0.220	0.188	0.225	-0.048
Window 1s	0.196	0.177	0.650	0.060	0.217	0.186	0.237	0.051
Window 2s	0.191	0.167	0.668	0.100	**0.211**	**0.181**	**0.268**	0.085
Window 3s	0.191	0.169	0.668	0.114	0.220	0.189	0.250	0.085
Window 4s	0.188	0.163	**0.681**	0.179	0.212	**0.180**	0.264	**0.110**
Window 5s	**0.187**	**0.162**	**0.681**	**0.201**	0.220	0.186	0.227	0.101

4.4. Extra labels

The next step I took was to exploit some of the dependency between the valence and arousal axis as suggested by Eerola and Vuoskoski [2010]. It has been reported that a hierarchical model where the predicted label for one of the axes is included in the feature vector for the other axis (i.e. the valence label in the feature vector for arousal prediction and the arousal label for valence prediction) can improve the accuracy of the model both in emotion recognition in music [Schmidt et al., 2010] and affect prediction from human behavior [Nicolaou et al., 2011]. In my experiments (see Table 4.4), the inclusion of the ground truth label for the other axis in the feature vector had a strong positive effect on valence prediction, but no effect on arousal prediction—results that agree with the findings in the literature [Schmidt et al., 2010].

Table 4.4: Results for the presence of the extra label in the feature vector, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
Basic	0.206	0.188	0.610	0.014	0.217	0.189	0.224	0.048
Extra label	0.207	0.186	0.606	0.028	0.209	0.182	0.272	0.061

4.5. Diagonal feature representation

Expectancy is another important factor to consider. There is a theory that violation of or conformity to expectancy when listening to music is a (main) source of mu-

sical emotion. It has been at least partially proven across different fields concerned with emotion in music (e.g. neuroimaging [Koelsch et al., 2010], experimental aesthetics [Hargreaves and North, 2010]). This has already been partially addressed by constructing a feature vector that includes features from the upcoming frames of a song (future window). Another way to address expectancy is to look directly at the change in feature values from the previous samples.

This takes us to the diagonal feature representation where in addition to the actual feature value, I also include the difference between the current value and the feature value in the previous sample, thus doubling the size of the feature vector. As we can see from Table 4.5, there is a small but consistent improvement present in all the evaluation metrics used, apart from short correlation for valence. Again, the difference in the arousal results is much more substantial than that for valence. While the change in representation does not have as much of an effect as expected, it definitely shows that the approach is a step in the right direction.

Table 4.5: Results for the diagonal feature vector representation, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
Basic	0.206	0.188	0.610	0.014	0.217	0.189	0.224	0.048
Diagonal	0.201	0.182	0.633	0.018	0.216	0.187	0.230	0.042

4.6. Moving relative feature representation

As the diagonal feature representation (Section 4.5) led to an improvement in the performance of the model, I extended that idea by combining it with the results described in the paper by Schubert [2004]. That paper showed that changes in different musical features of a piece take different amounts of time to affect the mood of a song. As the effect can sometimes take a number of seconds to take place, I decided to experiment with changing the span of features that get compared to the current sample in the feature vector. Instead of taking the difference between the current sample and the previous sample as in the diagonal feature representation, I first take the average of a feature over several previous samples and then take the difference between that and the current value. Each feature value then gets represented by two numbers—that moving average, taken across several previous samples, and the relative feature value, the difference between the moving average and the current value, doubling the size of the feature vector just like in the diagonal feature representation.

Table 4.6 shows the results of the moving relative feature representation, comparing it to the diagonal feature representation. Several things become apparent—first of all, the new representation is clearly beneficial to the model, as the improvement varies from small to large in the different metrics, but is always present. We can once again see a difference between the two axes—the model for the arousal axis is improved a lot more than that for the valence axis. Another thing is the improvement of the short correlation—while some of the other feature representations manage to improve all the other metrics, similarly to the time delay representation,

moving averages have a large positive effect on the short correlation in addition to a positive effect on the other metrics.

Table 4.6: Results for the moving relative feature vector representation, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
Diagonal	0.201	0.182	0.633	0.018	0.216	0.187	0.230	0.042
Mov. rel. 2s	0.191	0.173	0.664	0.116	0.216	0.187	0.236	0.100
Mov. rel. 3s	0.189	0.169	0.673	0.146	0.212	0.182	0.265	0.112
Mov. rel. 4s	0.188	0.168	0.677	0.177	0.213	0.183	**0.266**	0.140
Mov. rel. 5s	**0.186**	**0.165**	**0.682**	**0.199**	**0.212**	**0.180**	**0.266**	**0.144**

4.7. Relative feature representation

Taking the idea of the moving relative feature representation one step further, I introduced relative feature representation (papers Imbrasaitė and Robinson [2013] and Imbrasaitė et al. [2013]). In the relative feature representation each feature is represented, once again, by two values. This time the average is taken over the whole extract (or song) and then the difference is taken between that average and the current value. This way, the focus is not simply on the temporal change between now and the near past, but on the difference between the general mood of the song and the current sample. Expectancy is still the driving idea behind this feature representation, but now we are looking at a more global expectation of a listener.

To decouple the effect that including the average for each feature over a song might have to the results, I first tested a representation where only the average for each feature is added to the basic feature representation, not changing the actual feature value. This, when tested against the baseline method, had absolutely no effect on the results.

When relative feature representation is used (replacing the actual feature values with their differences from the overall average), the model shows great improvement over the baseline method. It is noticeable in the model for the valence axis, where there is a small improvement for long RMSE, but a large improvement for both short RMSE (7%) and long correlation (13%). The improvement is even larger in the model for the arousal axis—every metric is improved from the long RMSE and correlation (both by 11%) to short RMSE where the improvement is 17%.

Table 4.7: Results for the relative feature vector representation, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
Basic	0.206	0.188	0.610	0.014	0.217	0.189	0.224	0.048
Relative	0.183	0.153	0.697	0.054	0.212	0.176	0.274	0.014

The obvious next step is to combine the relative feature representation with the time delay/future window (Section 4.3). As the overall average for each feature

that is included in the relative feature representation stays the same in all the future and delay frames, only the difference between that and the delay/future value is included, therefore adding 50% of the original feature vector size with every added second.

Table 4.8: Results for the time delay/future window in relative feature representation compared with the best results of basic feature representation, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
Relative	0.183	**0.153**	0.697	0.054	**0.212**	**0.176**	**0.274**	0.014
Basic d. (5s)	0.187	0.166	0.679	**0.198**	0.214	0.182	0.271	**0.193**
Delay 1s	0.184	0.155	0.694	0.037	0.214	0.177	0.262	-0.038
Delay 2s	0.183	0.156	0.701	0.050	0.215	**0.176**	0.254	-0.071
Delay 3s	**0.182**	0.155	**0.704**	0.012	0.213	0.177	0.270	-0.099
Delay 4s	**0.182**	0.156	0.699	0.006	0.219	0.180	0.270	-0.091
Delay 5s	**0.182**	0.157	0.703	-0.006	0.215	0.180	0.266	-0.113
Basic f. (4s)	0.199	0.177	0.639	-0.040	0.218	0.190	0.234	-0.021
Future 1s	0.184	0.154	0.696	0.055	**0.212**	**0.174**	0.264	0.013
Future 2s	0.181	**0.153**	0.705	0.087	0.214	0.177	0.272	0.037
Future 3s	0.181	**0.153**	0.708	0.097	0.220	0.180	0.254	0.039
Future 4s	**0.179**	**0.153**	**0.709**	0.126	0.217	0.177	0.254	0.110
Future 5s	0.180	0.154	0.707	**0.129**	0.218	0.178	0.245	0.063
Basic w. (4s)	0.188	0.163	0.681	**0.179**	**0.212**	0.180	0.264	0.110
Window 1s	0.183	**0.154**	0.698	0.050	0.220	0.180	0.220	-0.008
Window 2s	0.183	0.155	**0.701**	0.063	0.214	**0.176**	0.257	-0.044
Window 3s	0.183	0.157	0.696	0.069	0.216	0.179	0.249	-0.080
Window 4s	0.184	0.159	0.689	0.051	0.217	0.184	0.237	-0.029
Window 5s	0.183	0.158	0.693	0.057	0.217	0.182	0.239	-0.026

Table 4.8 shows the results achieved with this feature representation, comparing them with both the standalone relative feature representation and the best results achieved with delay/future windows applied to the basic feature representation. There are several interesting trends visible from the results, especially when compared with the same approach applied to the basic feature representation (see Table 4.3). First of all, we see no improvement when adding features from the past samples—there is little consistency in the results, although they seem to deteriorate as the size of the vector is increased. This effect is especially visible through the short correlation metric, which is completely opposite to the effect we saw with the basic feature representation. Second of all, we see a completely opposite effect with the future window—we see a small improvement for the arousal axis (but not for valence), and we can once again see an improvement for the short correlation, contrary to what we saw with the basic feature representation. Finally, combining relative representation with the window samples has a similar effect to that of combining it with the delay samples—no improvement for most metrics and a negative effect on short correlation, especially for the valence axis. Overall, we can see that combining delay/future information with the relative feature representation gives no consistent improvement, and while it is still better than the

basic feature representation for most metrics, it cannot match the improvement for the short correlation.

Combination of the diagonal and relative feature representations (a basic concatenation of the two feature vectors) fails to lead to an improvement—Table 4.9 shows that while the results for the arousal axis are roughly the same for both the relative and joint representation, the valence axis' results are worse than those achieved by both the relative and the diagonal representation.

Table 4.9: Results for the joint diagonal and relative feature vector representation, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
Diagonal	0.201	0.182	0.633	0.018	0.216	0.187	0.230	**0.042**
Relative	0.183	**0.153**	0.697	**0.054**	**0.212**	**0.176**	**0.274**	0.014
Joint	**0.182**	0.154	**0.702**	0.035	0.221	0.182	0.249	-0.008

4.8. Multiple time spans

The final feature representation that I investigated is again based on the idea described in the paper by Schubert [2004]. It also relies on the moving relative feature representation (Section 4.7) as its starting point. In this feature representation, each feature value is represented by several averages taken over multiple time spans—starting from a 1 second time span (just the current value) and going all the way to an average taken over 5 seconds (the longest timespan considered in my experiments)—so a feature vector that is 5 times bigger than the original.

The results achieved with this feature representation (Table 4.10) unsurprisingly resemble those achieved by the moving relative feature representation (Section 4.6). They show a large improvement over the basic model, but not over the relative feature representation or even the moving relative feature representation with 5 s average. The only metric in which this representation is substantially better than the relative representation is short correlation. Unfortunately, especially given the much larger vector size, this feature representation does not look like a promising option.

Table 4.10: Results for the multiple time spans feature vector representation, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
Basic	0.206	0.188	0.610	0.014	0.217	0.189	0.224	0.048
Relative	**0.183**	**0.153**	**0.697**	0.054	0.212	**0.176**	**0.274**	0.014
Mov. rel. 5s	0.186	0.165	0.682	**0.199**	0.212	0.180	0.266	**0.144**
Time spans	0.186	0.165	0.685	0.129	**0.210**	0.180	**0.272**	0.108

4.9. Discussion

I believe that an important observation can be made from the results of these experiments—we see different levels of improvement that different feature representations have on the valence and arousal models. That seems to imply that in order to achieve the best results, different feature representations and/or feature fusion techniques might need to be used for the two axes.

Another important observation is that different feature representations have effects that are observable with different evaluation metrics. It confirms the ideas presented in the previous chapter—most importantly, it means that the best feature representation depends on the particular goals of the system (represented by an evaluation metric), and cannot be chosen universally. The most appropriate feature representation might also depend on the length of a song—relative feature representation, for example, might be more suitable for shorter songs, as it achieves higher long correlation and lower long RMSE, while moving relative representation might be more suitable for longer songs, as it achieves higher short correlation.

Moreover, it is clear from the improvements brought by the feature representations that the bag-of-frames approach is generally insufficient and can be improved by including some of the temporal information lost by dividing the signal into individual frames. Some of the feature representations, relative feature representation and delay window especially, also confirm the idea of expectancy and its effect on musical emotion.

4.10. Conclusions

In this chapter I have described the most popular features that are used for emotion recognition in music, as well as the main tools used for extracting them and the datasets that I am using in this dissertation.

The main contribution of this chapter is the 5 novel ways of representing feature vectors used in the machine learning solutions for this task. All 5 representations (time window, diagonal representation, moving relative representation, relative representation and multiple time spans representation) are based on findings in musicology and other fields of Music and Emotion and greatly improve the performance of a continuous dimensional music emotion recognition system trained with an SVR model, when compared to the standard feature representation.

Delay window and moving relative feature representations achieve the highest short correlation (0.198 and 0.199 for arousal and 0.193 and 0.144 for valence, respectively), while relative representation achieves the best results as measured by the other 3 metrics (11.2% reduction in long RMSE, 18.6% reduction in short RMSE and 14.3% increase in long correlation when compared to the basic representation).

Machine learning models

5

Most of the work on continuous emotion prediction in music uses the bag-of-frames approach—each frame (usually each second) is considered separately and used as a separate example for the machine learning method used. While some of the temporal information can be re-encoded into the feature vector (some examples of which are described in Chapter 4), the learning process is completely unaware of any relationships between the different samples. In this chapter I describe Support Vector Regression (SVR)—a machine learning method that is often used (here and elsewhere) as a baseline method. SVR is well suited for the bag-of-frames approach to dimensional emotion recognition and tracking. In addition to that, I also describe two other machine learning models—Continuous Conditional Random Fields (CCRF) and Continuous Conditional Neural Fields (CCNF)—that allow the temporal relationship between the different samples to be encoded and used during the training and prediction—these models have never been used for emotion tracking in music before. I provide a comparison between the three models, and show how their performance depends on the particular datasets that are used, as well as various feature representation techniques that I used.

Some of the work described in this chapter has been done in collaboration with Tadas Baltrušaitis, who worked on developing the CCRF and CCNF machine learning methods. That, and the results of some of other experiments are published in:

Emotion tracking in music using continuous conditional random fields and baseline feature representation, Vaiva Imbrasaitė, Tadas Baltrušaitis, Peter Robinson, *AAM workshop, IEEE International Conference on Multimedia, San Jose, CA, July 2013*

CCNF for continuous emotion tracking in music: comparison with CCRF and relative feature representation, Vaiva Imbrasaitė, Tadas Baltrušaitis, Peter Robinson, *MAC workshop, IEEE International Conference on Multimedia, Chengdu, China, July 2014*

Music emotion tracking with Continuous Conditional Neural Fields and Relative Representation, Vaiva Imbrasaitė, Peter Robinson, *The Mediaeval 2014 task: Emotion in music. Barcelona, Spain, October 2014*

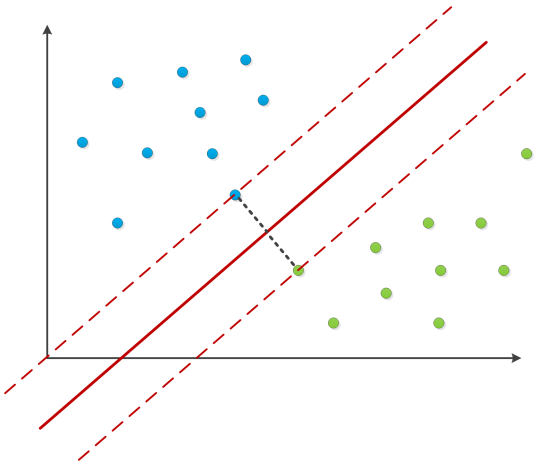


Figure 5.1: A diagram depicting an example of linear classification. The dashed lines represent the maximum margin lines with the solid red line representing the separation line.

5.1. Support Vector Regression

Support Vector Machines (SVM, also called Support Vector Networks) are a machine learning model primarily used for classification. Classification is achieved by representing the data points of interest as vectors in a high-dimensional or infinite-dimensional space and finding a hyperplane that best separates the vectors that belong to different classes. It was first introduced by Cortes and Vapnik [1995] and has been widely used ever since. A couple of years later, an extension of SVM was introduced by Drucker et al. [1996] that is able to cope with regression–Support Vector Regression (SVR), which is the model used in my work. As SVR is essentially an extension of SVM, the following sections will describe the SVM model first, then its kernels, and finally I will explain the difference between SVR and SVM.

All of the experiments described in this thesis that use SVR rely on a popular SVR implementation by Chang and Lin [2001]—LIBSVM.

5.1.1. Model description

The easiest and most straight-forward form of machine learning is linear classification. In the basic case we have a two-dimensional space with points belonging to two different classes and we are trying to find a line that best separates the two classes (see Figure 5.1). Ideally, we want to find a line such that all the points from one class lie on one side of the line, and all the points from the other class lie on the other side of the line. Moreover, to achieve the most generalizable solution, we want the line to be as far away from the two closest points of the two classes as possible, i.e. we want the margin to be as big as possible. In the more general case of linear SVM, we view each data point as a p-dimensional vector (where p is the size of the feature vector, or the number of numbers that describe each sample point) and we are looking for a $(p-1)$-dimensional hyperplane to separate the vectors.

As the number of dimensions can be very big or even infinite, we need to make sure that the construction and the representation of the separating hyperplane remains computable. This is achieved by only using a small amount of training data to determine the margin—the support vectors. When only a fraction of the training data is used to build the final solution, it increases the generalizability of the model. The expected probability of classification error is bounded by:

$$E(Pr(error)) \leq \frac{E(\text{number of support vectors})}{\text{number of training vectors}} \tag{5.1}$$

Since the bound (Equation 5.1) does not explicitly refer to the number of dimensions in the separating hyperplane, a well-generalizable model can be constructed with a small number of vectors even in an infinite space.

For a set of training examples $(y_1, \mathbf{x}_1), \ldots (y_n, \mathbf{x}_n)$, where $y_i \in \{-1, 1\}$, such a hyperplane can be defined as:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \; i = 1, \ldots, n \tag{5.2}$$

where $\mathbf{w}$ is a vector and b is a scalar used to define the hyperplane. The optimal hyperplane, achieving the largest margin, is then defined as:

$$\mathbf{w_0} \cdot \mathbf{x} + b_0 = 0 \tag{5.3}$$

The weights $\mathbf{w}_0$ that define the optimal hyperplane can also be written as a linear combination of the support vectors:

$$\mathbf{w}_0 = \sum_{i \in \text{support vectors}} y_i \alpha_i^0 \mathbf{x}_i \tag{5.4}$$

It can be shown that $\alpha > 0$ only for support vectors (see the Appendix of the paper by Smola and Schölkopf [2004]), simplifying the equation above and allowing us to represent all the weights by $\mathbf{\Lambda}_0^T = (\alpha_1^0, \ldots, \alpha_n^0)$.

If the training set can be separated by a hyperplane without an error, then the weights can be found by solving a quadratic programming problem:

$$W(\mathbf{\Lambda}) = \mathbf{\Lambda}^T \mathbf{1} - \frac{1}{2} \mathbf{\Lambda}^T \mathbf{D} \mathbf{\Lambda} \tag{5.5}$$

subject to

$$\mathbf{\Lambda} \geq \mathbf{0} \tag{5.6}$$

$$\mathbf{\Lambda^T Y} = 0 \tag{5.7}$$

Where $\mathbf{1}^T = (1, \ldots, 1)$ is a n-dimensional vector, $\mathbf{Y}^T = (y_1, \ldots, y_n)$ is the n-dimensional vector of training labels, and $\mathbf{D}$ is a symmetric $n \times n$-matrix with elements $D_{ij} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$.

Now $\mathbf{\Lambda}_0$, if it exists, can be found through the following iterative process. We start off by dividing the whole training set into sets with manageable number of

training vectors in each. We solve the quadratic problem maximising (5.5) under constraints (5.6) and (5.7). Either we are not able to find $\mathbf{\Lambda}_1$, in which case we cannot find a hyperplane that separates the training set without errors in the whole training set, or the optimal hyperplane for the first set of data is found. The zero coefficients present in $\mathbf{\Lambda}_1$ correspond to non-supporting vectors, and so they can be discarded, while the vectors with non-zero coefficients can now be included into the second set. We can also discard all the vectors in the second set that satisfy the constraint (5.2) using $\mathbf{\Lambda}_1$ as $\mathbf{w}$, as they are already taken care of by the current supporting vectors, and the process gets repeated with the new second set to find $\mathbf{\Lambda}_2$. Continuing the process incrementally we achieve $\mathbf{\Lambda}_* = \mathbf{\Lambda}_0$.

Soft margin hyperplane

Unfortunately, not all sets of training vectors can be linearly separated by a hyperplane, often due to the noisiness of the training data. Cortes and Vapnik [1995] offer a solution—soft margin hyperplane. A soft margin hyperplane allows vectors to fall on the other side of the margin, but still enables us to find the optimal hyperplane with the largest margin and the smallest error.

We define these errors as ξ_i, such that

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \; i = 1, \ldots, n \tag{5.8}$$

$$\xi_i \geq 0, \; i = 1, \ldots, n \tag{5.9}$$

and we aim to minimise $\sum_{i=1}^{n} \xi_i$. If the vectors with non-zero ξ_i were excluded from the training set, we could once again find the optimal separating hyperplane without any errors. This idea of soft margin hyperplane can be expressed by: minimising the functional

$$\frac{1}{2}\mathbf{w}^2 + CF(\sum_{i=1}^{n} \xi_i) \tag{5.10}$$

with the constraints (5.8) and (5.9), and where $F(u)$ is a monotonic convex function and C is a constant.

The quadratic programming problem (5.5) for the optimal solution can now be rewritten as:

$$W(\mathbf{\Lambda}) = \mathbf{\Lambda}^T\mathbf{1} - \frac{1}{2}(\mathbf{\Lambda}^T\mathbf{D}\mathbf{\Lambda} + \frac{\alpha_{max}^2}{C}) \tag{5.11}$$

subject to the same constraints (5.6 and 5.7) (see the Appendix in the paper by Smola and Schölkopf [2004] for more details). Unfortunately, due to the additional term this problem is no longer quadratic, but it now belongs to a group of so-called convex programming problems. It can therefore be solved as an n-dimensional convex problem or it can be rewritten to include another cost-related parameter and be turned into a dual-quadratic programming problem therefore requiring a $(n+1)$-dimensional solution.

For more details on the derivations and the solutions to the quadratic problems see the original paper by Cortes and Vapnik [1995] or the tutorial by Smola and Schölkopf [2004].

Sequential Minimal Optimization

Training an SVM that uses soft margins can be extremely time consuming which historically limited the use of SVMs until the Sequential Minimal Optimization (SMO) was introduced by Platt [1998]. The use of SMO sped up the training process by several orders of magnitude and the optimisation is now widely used and included in the LIBSVM implementation used in my work.

SMO avoids using numerical quadratic programing solutions, and instead solves the problem through an iterative process of choosing the smallest possible optimisation problems at every step and solving them analytically. The smallest possible optimisation problem in SVM training involves two Lagrange multipliers—SMO chooses two such multipliers, analytically finds their optimal values and then updates the SVM to reflect the new optimal values. While this process involves solving more sub-problems than the standard quadratic programming techniques, each individual sub-problem is much smaller and can be solved much faster, therefore improving the performance of the whole algorithm. It also does not require any additional matrix storage making it more feasible for personal computers as well as minimising the risk of numerical precision problems. The SMO algorithm consists of two steps that get repeated until the problem is solved: choosing of the next pair of Lagrange multipliers to solve (using a heuristic) and the analytical method for solving them.

Support Vector Regression

SVR (introduced by Drucker et al. [1996] and well explained in a tutorial by Smola and Schölkopf [2004]) is based on the same idea as SVM except now instead of searching for a hyperplane to separate two classes of vectors we are looking for a function to describe a set of vectors. In the basic case we want to find a function $f(\mathbf{x})$ which for all the training samples $\mathbf{x}_i$ produces a result that deviates from the actual observation y_i by at most ε and stays as flat as possible. This can be expressed as a convex optimisation problem:

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}\|\mathbf{w}^2\| \\ &\text{subject to} && \begin{cases} y_i - \mathbf{w}\cdot\mathbf{x}_i - b \leq \varepsilon \\ \mathbf{w}\cdot\mathbf{x}_i + b - y_i \leq \varepsilon \end{cases} \end{aligned} \tag{5.12}$$

where b is a scalar and $\mathbf{w} \in \chi$, the input space $\mathbb{R}^d$. Just like in the SVM solution, $\mathbf{w}$ can be expressed as a sum of coefficients ($\mathbf{w} = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\mathbf{x}_i$). The solution, again, depends only on the training vectors and not on the dimensionality of the input or the feature space.

Now, just like with SVM, it might not always be possible to achieve this perfect solution, so we introduce slack variables ξ_i, ξ_i^* that are analogous to the soft margin approach to SVM, which is described in Section 5.1.1. We rewrite the SVR equation

(5.12) to:

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}\|\mathbf{w}^2\| + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ &\text{subject to} && \begin{cases} y_i - \mathbf{w}\cdot\mathbf{x}_i - b \leq \varepsilon + \xi_i \\ \mathbf{w}\cdot\mathbf{x}_i + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \tag{5.13}$$

where $C > 0$ is the cost coefficient which determines the trade-off between the flatness of f and the deviation from ε. It is important to note that we only care about deviation from f if it is greater than ε, and consider it to be 0 if it is not. The dual form of the optimisation problem (5.13) can then be solved using Lagrange multipliers.

As part of the solution to this optimisation problem we need to define the Karush-Kuhn-Tucker conditions:

$$\begin{aligned} \alpha_i(\varepsilon + \xi_i - y_i + \mathbf{w}\cdot\mathbf{x}_i + b) = 0 \\ \alpha_i^*(\varepsilon + \xi_i^* + y_i - \mathbf{w}\cdot\mathbf{x}_i - b) = 0 \end{aligned} \tag{5.14}$$

and

$$\begin{aligned} (C - \alpha_i)\xi_i = 0 \\ (C - \alpha_i^*)\xi_i^* = 0 \end{aligned} \tag{5.15}$$

From the conditions (5.14) and (5.15) it follows that for all $\mathbf{x}_i$ such that $|f(\mathbf{x}_i) - y_i| < \varepsilon$ the second factor in (5.14) is non-zero, which means that α_i, α_i^* has to be zero. This in turn means that, just like in the SVM case, the solution depends on only a subset of the training data—a handful of support vectors which have non-zero α_i, α_i^* coefficients.

5.1.2. Kernels

Kernels make SVM or SVR more powerful than simple linear classification or regression. It is not a surprising that the points in the two classes can sometimes be difficult to separate with a single hyperplane. In a more general case of SVM or SVR, we use a kernel function $k(x, y)$ to transform the given p-dimensions into a higher- or even an infinite-dimensional space, where the separation might be easier. To ensure reasonable computation costs, the kernel functions are designed in such a way that the dot product can be computed easily. This changes the problem from linear classification to non-linear classification (see Image 5.2 for an illustration).

The most common kernels used in SVM or SVR implementations, in addition to the linear kernel, are:

- Polynomial kernel: $k(x_i, x_j) = (\gamma x_i x_j + r)^d$, $\gamma > 0$
- Gaussian Radial Basis kernel (RBF): $k(x_i, x_j) = exp(-\gamma(x_i - x_j)^2)$, for $\gamma > 0$, sometimes parametrized using $\gamma = \frac{1}{2}s^2$
- Sigmoid kernel: $k(x_i, x_j) = tanh(\gamma x_i^T x_j + r)$

The optimal choice of a kernel and of its parameters will depend on a particular problem and the training set, so care must be taken when choosing them.

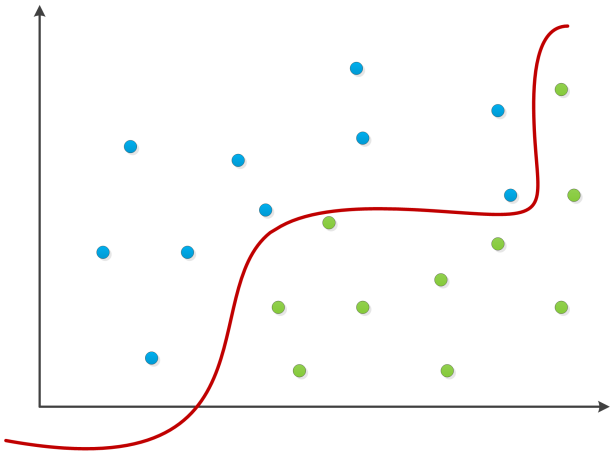


Figure 5.2: A diagram depicting an example of non-linear classification. The solid red line represents the best non-linear separation line, while a linear separation without any errors is not possible in this example.

5.1.3. Comparison

A lot of research that uses SVR as a baseline to compare the new approaches with, tends to use Linear kernels as they are much more straightforward, have fewer parameters and train faster than the other kernels that can be used. Unfortunately, their simplicity can also lead to poorer results, so the superior results achieved by the new algorithms have to be viewed with caution, unless the experimental method justifies a simplified approach.

To find out which kernel is the most appropriate for my work (as the correct choice of kernel is problem- and feature-vector dependent), I tested all 4 kernels provided in the LIBSVM implementation with both the standard and relative feature representation. I chose to use the standard feature representation as that corresponds to the work done by other researchers in the field, and the relative representation, as it was one of the best performing feature vector representation (as described in Chapter 4). The experimental conditions were kept the same as elsewhere in the dissertation, and their full description can be found below, in Section 5.4.1. The only difference here is the selection of the parameters—the linear kernel requires only one parameter C which is chosen using cross-validation; g and C for RBF kernel are chosen through grid-search and cross-validation, as explained in Section 5.4.1; the polynomial kernel and the sigmoid kernels have an extra coefficient—the offset—that is kept at 0, as per default (making the training process for the sigmoid kernel identical to RBF kernel) and also a maximum degree for polynomial that is chosen through an extra dimension in the grid-search.

The results in Table 5.1 show that there is some variation in performance between the different kernels available. It is clear that choosing to use the linear kernel (mostly due to its simplicity, its ease of use and short training time) or the sigmoid kernel can lead to inferior results, which are especially harmful and deceiving when such a method is used as the baseline method for comparison with other work. The results show that the RBF kernel and the polynomial kernels achieve

Table 5.1: Results for the 4 different kernels used in SVR, basic and relative (R) feature representation, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
Linear	0.210	0.191	0.598	-0.005	0.224	0.196	0.200	0.030
Poly	0.207	0.189	0.607	0.012	0.220	0.193	**0.217**	0.038
RBF	**0.206**	**0.188**	**0.611**	**0.014**	**0.217**	**0.190**	**0.218**	**0.050**
Sigmoid	0.211	0.191	0.594	-0.003	0.222	0.194	0.179	0.030
Linear-R	0.189	0.159	0.681	0.039	0.233	0.189	0.241	0.006
Poly-R	**0.182**	**0.153**	**0.706**	0.040	0.225	0.185	0.241	0.008
RBF-R	0.184	0.155	0.694	**0.045**	**0.211**	**0.174**	**0.243**	**0.020**
Sigmoid-R	0.198	0.168	0.670	0.025	0.219	0.180	0.242	0.011

similar performance in the continuous dimensional emotion prediction task (with RBF showing a small advantage), at least with the features in hand. For my experiments I choose to use the RBF kernel, as it has fewer hyper-parameters that need to be picked than the polynomial kernel and therefore is less susceptible to overfitting, especially with smaller datasets.

Table 5.2: Results achieved with different training parameters values with the linear kernel in SVR, basic feature representation, standard and short metrics

C	Arousal				Valence			
	RMS	RMSs	Corr	Corrs	RMS	RMSs	Corr	Corrs
0.00195313	0.282	0.247	0.528	0.024	0.236	0.197	0.141	0.042
0.00390625	0.255	0.227	0.539	0.024	0.232	0.194	0.165	0.043
0.0078125	0.234	0.211	0.555	0.023	0.227	0.191	0.179	0.042
0.015625	0.221	0.200	0.573	0.024	0.222	0.189	0.210	0.043
0.03125	0.213	0.193	0.592	0.023	0.218	0.187	0.226	0.044
0.0625	0.208	0.189	0.605	0.018	0.215	**0.185**	**0.245**	0.045
0.125	0.206	**0.187**	0.611	0.016	**0.214**	**0.185**	0.239	0.046
0.25	**0.205**	**0.187**	**0.614**	0.016	**0.214**	0.186	0.240	0.046
0.5	0.206	0.188	0.611	0.020	**0.214**	0.188	0.238	0.047
1	0.209	0.191	0.602	0.025	0.216	0.190	0.238	0.049
2	0.211	0.194	0.594	0.028	0.219	0.195	0.230	0.054
4	0.214	0.198	0.585	0.028	0.224	0.201	0.215	**0.051**
8	0.221	0.206	0.562	**0.030**	0.230	0.209	0.198	**0.051**

For the sake of interest, and to emphasise the point that not only the choice of kernels is important, but also the choice of correct training parameters, Table 5.2 shows the results of grid-search (with one parameter only) achieved with different values of the training parameter C. It is clear that the choice of such a parameter can have a dramatic effect on the results achieved by a model (with results deteriorating on either side of 0.125-0.25), so a lot of care must be taken when choosing it, as well as to avoid overfitting, which would make a model less generalisable.

5.2. Continuous Conditional Random Fields

As discussed previously in Chapter 4, the bag-of-frames approach ignores all of the temporal information present in the data. Even when some of the temporal information can be re-encoded into the feature vector, the standard machine learning methods (such as SVR) would still be unaware of any relationship between different samples and the predictions assigned to them. In other words, since emotion has temporal properties and is not instantaneous, we would like to explicitly model the temporal relationships between each time step. A recent and promising approach that would allow us to model such temporal relationships is Continuous Conditional Random Fields (CCRF) developed by Qin et al. [2008]. It is an extension of the classic Conditional Random Fields [Lafferty et al., 2001] to the continuous case. Furthermore, it has recently been extended by Baltrušaitis et al. [2013] so it can be used for continuous emotion prediction incorporating temporal information—which is the model used and described here.

5.2.1. Model definition

Here, we are building a hierarchical model where we rely on predictions from another model (most likely using the bag-of-frames approach) as input and we learn the relationship between them with CCRF.

CCRF is an undirected graphical model where conditional probability $P(y|x)$ is modeled explicitly. It is a discriminative approach, which Sutton and Mccallum [2006] have shown to achieve promising results for sequence labeling and segmentation. This is in contrast to generative models where a joint distribution $P(y, x)$ is modeled instead. The graphical model that represents the CCRF for emotion prediction is shown in Figure 5.3. As can be seen from the figure, the CCRF model focuses on modeling the temporal relationship between the samples rather than extracting the pattern from the features themselves.

In the description I will use the following notation:

- $\{\mathbf{x}_1^{(q)}, \mathbf{x}_2^{(q)}, \ldots, \mathbf{x}_n^{(q)}\}$ is a set of observed input variables (in this case an SVR prediction), $\mathbf{x}_i^{(q)} \in \mathcal{R}^m$
- $\{y_1^{(q)}, y_2^{(q)}, \ldots, y_n^{(q)}\}$ is a set of output variables that we wish to predict, $y_i^{(q)} \in \mathcal{R}$
- n is the number of frames/time-steps in a sequence
- m is the number of predictors used (generally we just use one, but multiple predictions per modality can easily be used)
- q indicates the q^{th} sequence of interest; when there is no ambiguity, q is omitted for clarity.

The CCRF model for a particular sequence is a conditional probability distribution with the probability density function:

$$P(\mathbf{y}|\mathbf{X}) = \frac{\exp(\Psi)}{\int_{-\infty}^{\infty} \exp(\Psi) d\mathbf{y}} \tag{5.16}$$

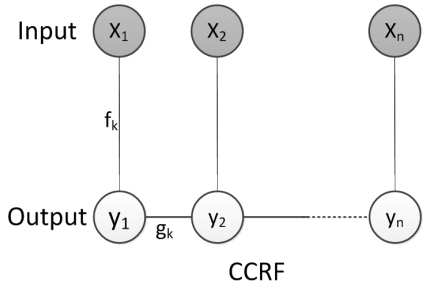


Figure 5.3: Graphical representation of the CCRF model. x_i represents the the i^{th} observation, and y_i is the unobserved variable we want to predict. Dashed lines represent the connection of observed to unobserved variables (f is the vertex feature). The solid lines show connections between the unobserved variables (edge features).

$$\Psi = \sum_i \sum_{k=1}^{m} \alpha_k f_k(y_i, \mathbf{X}) + \sum_{i,j} \beta g(y_i, y_j, \mathbf{X}) \tag{5.17}$$

Above, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ is the set of input feature vectors (can be represented as a matrix with per frame observations as rows), $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$ is the unobserved variable, or the label. $\int_{-\infty}^{\infty} \exp(\Psi) d\mathbf{y}$ is the normalisation (partition) function which makes the probability distribution a valid one (by normalising it to sum to 1). Following the convention of [Qin et al., 2008] we call f vertex features, and g edge features (a single vertex and a single edge feature is used in this model, so k is dropped in some subsequent equations). The model parameters α, and β would be provided for inference and need to be estimated during learning.

5.2.2. Feature functions

We define two types of features for the CCRF model, vertex features f_k and edge feature g.

$$f_k(y_i, \mathbf{X}) = -(y_i - \mathbf{X}_{i,k})^2, \tag{5.18}$$

$$g(y_i, y_j, \mathbf{X}) = -\frac{1}{2} S_{i,j} (y_i - y_j)^2. \tag{5.19}$$

Vertex features f_k represent the dependency between the $\mathbf{X}_{i,k}$ and y_i, for example dependency between a static emotion prediction from a regressor and the actual emotion label. Intuitively, the corresponding α_k for vertex feature f_k represents the reliability of that particular predictor. In this work only a single predictor is used, however, it is possible to use multiple regressors (see Baltrušaitis et al. [2013]).

Edge feature g represents the dependency between observations y_i and y_j, which described the relationship between the emotion prediction at time step j and the one at time step i. This is also affected by the similarity measure S—because we are using a fully connected model, the similarity S allows us to control the strength or existence of such connections. In this model the following similarity is used:

$$S_{i,j} = \begin{cases} 1, & |i-j| = 1 \\ 0, & \text{otherwise} \end{cases} \tag{5.20}$$

Thus we connect only the neighboring observations. The framework allows for easy creation of different similarity measures which could be appropriate for other applications.

The learning phase of CCRF will determine the parameters α and β. For example, it can learn that for one emotion neighbor similarities are more important than for others.

As in Radosavljevic et al. [2010], Qin et al. [2008] and Baltrušaitis et al. [2013], the feature function models the square error between prediction and a feature. Therefore the elements of the feature vector $\mathbf{x}_i$ should be predicting the unobserved variable y_i, for example, the Support Vector Regression predictions generated in the previously described experiments.

5.2.3. Learning

This section provides a description of how to estimate the parameters $\{\alpha, \beta\}$ of a CCRF with quadratic vertex and edge functions. We are given training data $\{\mathbf{x}^{(q)}, \mathbf{y}^{(q)}\}_{q=1}^{M}$ of M sequences, where each $\mathbf{x}^{(q)} = \{\mathbf{x}_1^{(q)}, \mathbf{x}_2^{(q)}, \ldots, \mathbf{x}_n^{(q)}\}$ is a sequence of inputs and each $\mathbf{y}^{(q)} = \{y_1^{(q)}, y_2^{(q)}, \ldots, y_n^{(q)}\}$ is a sequence of real valued outputs. We also use the matrix $\mathbf{X}$ to denote the concatenated sequence of inputs.

In learning, we want to pick the α and β values that optimise the conditional log-likelihood of the CCRF:

$$L(\alpha, \beta) = \sum_{q=1}^{M} \log P(\mathbf{y}^{(q)} | \mathbf{x}^{(q)}) \tag{5.21}$$

$$(\bar{\alpha}, \bar{\beta}) = \arg\max_{\alpha,\beta}(L(\alpha, \beta)) \tag{5.22}$$

As the problem is convex [Qin et al., 2008], the optimal parameter values can be determined using standard techniques such as stochastic gradient ascent, or other general optimisation techniques.

In order to guarantee that the partition function is integrable, it is constrained $\alpha > 0$ and $\beta > 0$ [Qin et al., 2008; Radosavljevic et al., 2010]. Such constrained optimisation can be achieved by using partial derivatives with respect to $\log \alpha$ and $\log \beta$ instead of just α and β. We also add a regularisation term in order to avoid overfitting. The regularisation is controlled by λ_α and λ_β hyper-parameters (determined during cross-validation):

$$\frac{\partial \log(P(\mathbf{y}|\mathbf{X}))}{\partial \log \alpha} = \alpha\left(\frac{\partial \log(P(\mathbf{y}|\mathbf{X}))}{\partial \alpha} - \lambda_\alpha \alpha\right) \tag{5.23}$$

$$\frac{\partial \log(P(\mathbf{y}|\mathbf{X}))}{\partial \log \beta} = \beta\left(\frac{\partial \log(P(\mathbf{y}|\mathbf{X}))}{\partial \beta} - \lambda_\beta \beta\right) \tag{5.24}$$

The full derivation and definition of the partial derivatives can be found in Baltrušaitis et al. [2013].

The full learning algorithm is described in Algorithm 1.

Algorithm 1 CCRF learning algorithm

Require: $\{\mathbf{X}^{(q)}, \mathbf{y}^{(q)}, S_q\}_{q=1}^{M}$
  Params: number of iterations T, learning rate ν, $\lambda_\alpha, \lambda_\beta$
  Initialise parameters $\{\alpha, \beta\}$
  for r = 1 **to** T **do**
    for i = 1 **to** N **do**
      Compute gradients of current query (Eqs.(5.23),(5.24))
      $\log \alpha = \log \alpha + \nu \frac{\partial \log(P(\mathbf{y}|\mathbf{X}))}{\partial \log \alpha}$
      $\log \beta = \log \beta + \nu \frac{\partial \log(P(\mathbf{y}|\mathbf{x}))}{\partial \log \beta}$
      Update $\{\alpha, \beta\}$
    end for
  end for
  return $\{\bar{\alpha}, \bar{\beta}\} = \{\alpha, \beta\}$

5.2.4. Inference

Because the CCRF model can be viewed as a multivariate Gaussian, inferring $\mathbf{y}$ values that maximise $P(\mathbf{y}|\mathbf{x})$ is straightforward. The prediction is the mean value of the distribution.

$$\mathbf{y}' = \arg\max_{\mathbf{y}}(P(\mathbf{y}|\mathbf{X})) \tag{5.25}$$

For more details on the inference algorithm please see Baltrušaitis et al. [2013].

5.3. Continuous Conditional Neural Fields

The major disadvantage of CCRF is that we have to train two machine learning models, and while the temporal information is made explicit in the second training process, it is not available throughout the whole training—essentially turning CCRF into a exceptionally intelligent smoothing function. The Continuous Conditional Neural Fields (CCNF) model is a novel regression model developed by Baltrušaitis et al. [2014] (shown in Figure 5.4) that combines the nonlinearity of Conditional Neural Fields [Peng et al., 2009] and the flexibility and the continu-

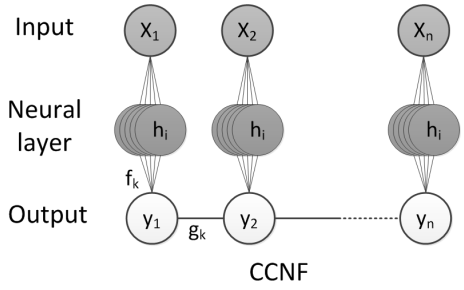


Figure 5.4: Linear-chain CCNF model. The input vector $\mathbf{x}_i$ is connected to the relevant output scalar y_i through the vertex features that combine the h_i neural layers (gate functions) and the vertex weights $\boldsymbol{\alpha}$. The outputs are further connected with edge features g_k

ous output of Continuous Conditional Random Fields [Qin et al., 2008]. It can learn non-linear dependencies between the input and the output and model the temporal and spatial relationship between the samples in a sequence, therefore is especially suitable for time-varying emotion prediction task.

5.3.1. Model definition

CCNF is an undirected graphical model that can model the conditional probability of a continuous valued vector $\mathbf{y}$ (for example the emotion in valence space) depending on continuous $\mathbf{x}$ (for example audio features).

The following notation is used below: $\boldsymbol{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ is a set of observed input variables, $\mathbf{X}$ is a matrix where the i^{th} column represents $\boldsymbol{x}_i$, $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$ is a set of output variables that we wish to predict, $\mathbf{x}_i \in \mathcal{R}^m$ and $y_i \in \mathcal{R}$ (patch expert response), n is the length of the sequence of interest.

The model for a particular set of observations is a conditional probability distribution with the probability density function:

$$P(\mathbf{y}|\boldsymbol{x}) = \frac{\exp(\Psi)}{\int_{-\infty}^{\infty} \exp(\Psi) d\mathbf{y}} \tag{5.26}$$

Above, $\int_{-\infty}^{\infty} \exp(\Psi) d\mathbf{y}$ is the partition function which forces the probability distribution to sum to 1.

We define two types of features in CCNF: vertex features f_k and edge features g_k. The potential function is defined as:

$$\Psi = \sum_{i} \sum_{k=1}^{K1} \alpha_k f_k(y_i, \mathbf{x}_i, \boldsymbol{\theta}_k) + \sum_{i,j} \sum_{k=1}^{K2} \beta_k g_k(y_i, y_j) \tag{5.27}$$

In order to guarantee that the partition function is integrable [Qin et al., 2008] we constrain $\alpha_k > 0$ and $\beta_k > 0$, while $\boldsymbol{\Theta}$ is unconstrained. The model parameters $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots \alpha_{K1}\}$, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots \boldsymbol{\theta}_{K1}\}$, and $\boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots \beta_{K2}\}$ are learned and used for inference during testing.

The vertex features f_k represent the mapping from the $\mathbf{x}_i$ to y_i through a one layer neural network, where $\boldsymbol{\theta}_k$ is the weight vector for a particular neuron k.

$$f_k(y_i, \mathbf{x}_i, \boldsymbol{\theta}_k) = -(y_i - h(\boldsymbol{\theta}_k, \mathbf{x}_i))^2 \tag{5.28}$$

$$h(\boldsymbol{\theta}, \mathbf{x}_i) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}_i}} \tag{5.29}$$

The number of vertex features $K1$ is determined experimentally during cross-validation, and in the experiments I tried $K1 = \{10, 20, 30\}$.

The edge features g_k represent the similarities between observations y_i and y_j. This is also affected by the neighborhood measure $S^{(k)}$, which allows us to control the existence of such connections.

$$g_k(y_i, y_j) = -\frac{1}{2} S_{i,j}^{(k)} (y_i - y_j)^2. \tag{5.30}$$

In the linear chain CCNF model, g_k enforces smoothness between neighboring nodes. I define a single edge feature, i.e. $K2 = 1$. I define $S^{(1)}$ to be 1 only when the two nodes i and j are neighbors in a chain, otherwise it is 0.

5.3.2. Learning and Inference

We are given training data $\{\boldsymbol{x}^{(q)}, \mathbf{y}^{(q)}\}_{q=1}^{M}$ of M song samples, together with their corresponding dimensional continuous emotion labels. The dimensions are trained separately. In this section I describe how to estimate the parameters $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Theta}\}$, given the training data. It is important to note that all of the parameters are optimised jointly.

In learning, we want to pick the $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\Theta}$ values that optimise the conditional log-likelihood of the model on the training sequences:

$$L(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Theta}) = \sum_{q=1}^{M} \log P(\mathbf{y}^{(q)} | \boldsymbol{x}^{(q)}) \tag{5.31}$$

$$(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}, \bar{\boldsymbol{\Theta}}) = \arg\max_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Theta}} (L(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Theta})) \tag{5.32}$$

Similarly to Baltrušaitis et al. [2013] and Radosavljevic et al. [2010], Equation 5.26 is turned into multivariate Gaussian form. It helps with the derivation of the partial derivatives of log-likelihood, and with the inference.

$$P(\mathbf{y}|\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{y} - \boldsymbol{\mu})), \tag{5.33}$$

$$\Sigma^{-1} = 2(A + B) \tag{5.34}$$

The diagonal matrix A represents the contribution of α terms (vertex features) to the covariance matrix, and the symmetric B represents the contribution of the β terms (edge features). They are defined as follows:

$$A = (\sum_{k=1}^{K1} \alpha_k) I \tag{5.35}$$

$$B_{i,j} = \begin{cases} (\sum_{k=1}^{K2} \beta_k \sum_{r=1}^{n} S_{i,r}^{(k)}) - (\sum_{k=1}^{K2} \beta_k S_{i,j}^{(k)}), & i = j \\ -\sum_{k=1}^{K2} \beta_k S_{i,j}^{(k)}, & i \neq j \end{cases} \tag{5.36}$$

We also define $\boldsymbol{\mu} = \Sigma \boldsymbol{d}$ which is the expected (mean) value of the Gaussian CCNF distribution.

$$\boldsymbol{\mu} = \Sigma \boldsymbol{d} \tag{5.37}$$

It is defined in terms of $\Sigma = (2A + 2B)^{-1}$, and a vector $\boldsymbol{d}$, that describes the linear terms in the Gaussian distribution:

$$\mathbf{d} = 2\alpha^T h(\Theta \mathbf{X}) \tag{5.38}$$

Above, Θ represents the combined neural network weights and $h(\Theta \mathbf{X})$, is an element-wise application of h on each element of the resulting matrix. Intuitively $\boldsymbol{d}$ is the contribution from the vertex features towards $\mathbf{y}$.

For learning we can use the constrained L-BFGS for finding locally optimal model parameters (with, e.g., the standard Matlab implementation of the algorithm). In order to make the optimisation both more accurate and faster, the partial derivatives of the $\log P(\mathbf{y}|\mathbf{x})$ are used, which are straightforward to derive and are similar to those of CCRF, see Section 5.2.3.

In order to avoid overfitting, L_2 norm regularisation terms are added to the likelihood function for each of the parameters types ($\lambda_\alpha ||\boldsymbol{\alpha}||_2^2$,$\lambda_\beta ||\boldsymbol{\beta}||_2^2$, $\lambda_\theta ||\boldsymbol{\Theta}||_2^2$). The values of $\lambda_\alpha, \lambda_\beta, \lambda_\Theta$ are determined during cross-validation, as is the number of neural layers.

For inference we need to find the value of $\mathbf{y}$ that maximises $P(\mathbf{y}|\boldsymbol{x})$. Because $P(\mathbf{y}|\boldsymbol{x})$ can be expressed as a multivariate Gaussian distribution (Equation 5.33) , the value of $\mathbf{y}$ that maximises it is the mean value of the distribution, hence:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}}(P(\mathbf{y}|\boldsymbol{x})) = \boldsymbol{\mu} \tag{5.39}$$

For more details on the derivations, see Baltrušaitis et al. [2014].

5.4. Comparison

To compare the suitability of the three machine learning methods (SVR, CCRF and CCNF) for the task of continuous dimensional emotion prediction, I tested them on several datasets and using several feature representations. The design of the experiment I used, described below, ensures that the conditions in which the models are tested are as fair as possible and that any differences that are apparent in the results are due to the differences in the performance and the power of the models and not the randomness present in the experiment or overfitting of the models.

5.4.1. Design of the experiments

The experimental design for this set of experiments is identical to that used in other experiments described in this dissertation, to allow a direct comparison between different approaches.

Several datasets were used to train the three models (see Section 2.4 for their full description). When the original MoodSwings dataset was used, only the non-EchoNest features provided were included in the feature vectors. Otherwise, the smaller MoodSwings dataset (where the audio files are available) with OpenSMILE extracted features was used. In addition to those, some of the experiments were repeated with the MediaEval 2014 development set, to see how the models performed with a much larger amount of data.

Features were averaged over a 1 s (or 0.5 s for MediaEval 2014) window and the average of the labels for that sample was used as the annotation. Each feature was normalised so that its maximum value over the whole feature set would be 1 and the minimum would be -1—that it is necessary for the machine learning models to be able to learn the importance of different features while ignoring the fact that their values might differ by several orders of magnitude. With all the machine learning models used, I trained two models separately—one for each axis.

Feature representation

Two types of feature representations are used in all of the experiments: basic and relative (see Chapter 4 in general and Section 4.7 in particular for more information). For both feature representations, the features are concatenated into either a single vector (feature-level fusion) or into 4 separate vectors for the original MoodSwings dataset: MFCC, chromagram, spectral contrast and SSD (model-level fusion) for the hierarchical, model-level fusion (described below). I also used several other (time-delay window and moving relative) feature representation techniques to compare SVR and CCNF, to see if they would have a similar effect with a more complicated model (see Chapter 4 for detailed description of the representations and their performance).

Cross-validation

5-fold cross-validation was used for all the experiments. The dataset was split into two parts—4/5 for training and 1/5 for testing, and this process was repeated 5 times. When splitting the dataset into folds, all of the feature vectors from a single song are put in the same fold. I chose to ignore the album and artist information, as I have shown previously (see Section 4.2.1) that the MoodSwings dataset used here does not suffer from the album effect. The distribution of songs over the folds was kept fixed throughout all the experiments, to have as little variation between the conditions as possible. The reported results were averaged over 5 folds.

For SVR-based experiments, 2-fold cross validation (splitting into equal parts) on the training dataset was used in each fold to choose the hyper-parameters. These were then used for training on the whole training dataset. Only the RBF kernel was used for the experiments, as it is the most appropriate kernel for this problem (see Section 5.1.3). As it has two hyper-parameters, a 2-dimensional grid-search is used with C ranging between 2^{-7} and 2^{3} at every power of 2, and g ranging between 2^{-9} and 2^{-1} at every power of 2.

The process for the CCRF-based experiments contained an extra step. The training dataset is split into two parts—one for SVR and one for CCRF, and we perform a 2-fold cross-validation on them individually to learn the hyper-parameters for each model in the same way as for the SVR-based experiments. The hyper-parameters for SVR are the same as those described in the paragraph above, while the CCRF requires two extra hyper-parameters λ_α and λ_β picked from the set {0.001, 0.01, 0.1, 1, 10, 100, 1000} using cross-validation.

For CCNF-based experiments 2-fold cross validation is again used to pick the hyper-parameters, but the results are averaged over 4 random seed initializations. The chosen hyper-parameters are used for training on the whole dataset—I randomly initialized the seed 20 times (using the best hyper-parameters) and picked the model with the highest likelihood (Equation 5.31) for testing. As there are 4 hyper-parameters that need to be picked, a 4-dimensional grid-search is done with the number of neural layers being picked from the set {10, 20, 30}, λ_α from {1, 10, 100}, and λ_β and λ_θ from {0.001, 0.01, 0.1, 1}.

It is important to note that the same folds were used for all of the experiments, and that the testing data was always kept separate from the training process in order to minimise the risk of overfitting.

5.4.2. Feature fusion

The CCNF model can accommodate different types of feature fusion. For feature fusion we just use single vector $\mathbf{x}_i$ containing all of the different modality features. For model-level fusion we split the vector into different modalities leading to more feature functions, where each modality has its own corresponding f_ks. We define separate vertex functions for MFCC, chromagram, spectral contrast and SSD features used in the original MoodSwings dataset.

For SVR-based model-level fusion, I trained 4 separate models—one for each class of features. Then a feature vector composed of the predicted labels only was used for the final model.

5.4.3. Results

Similarly to Chapter 4, all of the results in this chapter are reported using correlation and root-mean-square error (RMSE), both computed over a whole dataset, essentially concatenating all the songs into one (long) and per-song basis and then averaged (short). The short correlation that is reported is non-squared, so as not to hide any potential negative correlation. Chapter 3 explains the reasoning behind the use of both of these metrics as well as behind the difference between the two modes. Long metrics are reported purely for the purpose of easier comparison with other work described in the literature. For the same reason, some of the experiments also include the Euclidean distance as one of the metrics, as many of the papers using the MoodSwings dataset report Euclidean distance as one of the metrics. The average Euclidean distance is calculated as the distance between the two-dimensional position of the original label and the predicted labels in normal-

ized AV space (each axis normalized to span between 0 and 1). All metrics are calculated for each fold and the average over 5 folds is reported.

Please note that lower RMSE and Euclidean distance values correspond to better performance, while the opposite is true for correlation.

Standard feature representation

With the original MoodSwings dataset, CCNF with the basic feature representation consistently outperforms all of the other methods in all the evaluation metrics except for short correlation for valence, where CCRF performs better (Table 5.3). CCNF also outperforms the other two in the case for the Euclidean distance metric, where SVR achieves 0.128, CCRF–0.136 and CCNF–the average distance of 0.116. That and the long RMSE are the only metrics where SVR slightly outperforms CCRF, which otherwise is clearly the second best machine learning model for this problem. CCNF not only achieves better performance than the other two, but it can be seen that the results are substantially better than those of the other methods.

Since neural network-based models are particularly sensitive to the size of the feature vector used, we also tried to minimize the feature vector by omitting a class of features. As can be seen from Table 5.4, not including chromagram, octave-based spectral contrast or MFCC features in the feature vector improves the result even further (compare with Table 5.3), with the omission of either chromagram or the OBSC achieving similar results. It is also apparent from the results that there is a clear difference between the effect of these features on different axes—as expected, omission of MFCC has a negative effect on the arousal results, and the omission of chromagram or the OBSC is detrimental to the model for the valence axis. Based on the Euclidean metric, which combined both axes, there is virtually no difference between the three sets of features, as the omission of chromagram or OBSC achieves 0.116 and the omission of MFCC achieves the average distance of 0.117.

Table 5.3: Results comparing the CCNF approach to the CCRF and SVR with RBF kernel using basic feature vector representation on the original MoodSwings dataset.

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
SVR	0.194	0.178	0.645	0.011	0.220	0.186	0.211	0.007
CCRF	0.204	0.176	0.721	0.049	0.223	0.183	0.247	**0.090**
CCNF	**0.166**	**0.143**	**0.739**	**0.072**	**0.205**	**0.170**	**0.301**	0.019

Table 5.4: Results of CCNF with smaller feature vectors, same conditions as Table 5.3.

	Arousal				Valence			
W/o	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
Chroma	0.167	0.144	0.737	**0.068**	0.207	0.172	0.285	0.046
OSBC	**0.164**	**0.143**	**0.743**	0.047	0.208	0.169	0.285	0.040
MFCC	0.175	0.150	0.707	0.032	**0.200**	**0.164**	**0.315**	**0.089**

Table 5.5: Results comparing CCNF, CCRF and SVR with RBF kernels using relative feature representation on the original MoodSwings dataset.

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
SVR	**0.167**	**0.143**	**0.735**	0.046	0.209	0.170	**0.297**	0.035
CCRF	0.179	0.153	0.718	**0.071**	0.216	0.176	0.257	0.049
CCNF	**0.167**	0.145	0.733	0.058	**0.207**	**0.169**	0.281	**0.064**

Relative feature representation

The results on the original MoodSwings dataset with the relative feature representation are less consistent (Table 5.5). Even though CCNF clearly outperforms CCRF and SVR with the standard representation, the results are nearly identical to those achieved by the SVR model with the relative feature representation. That is especially evident with long, rather than short, evaluation metrics. The Euclidean distance achieved with SVR is also smaller than that achieved by CCNF (0.117 compared to 0.120). It is also clear that both of these models outperform CCRF on most metrics, including Euclidean distance, where CCRF achieved an average distance of 0.123.

The relative feature representation vector is twice the size of the standard representation. In order to potentially alleviate the problem of long feature vectors that might be causing this lack of performance, I tried decreasing the number of features included, in the same way as described above. Unlike those with the standard feature representation, these experiments failed to improve the results in any substantial way.

Other datasets

Based on the results described above, it seems that CCRF does not seem to give interesting results—when a simple feature representation technique is used, it improves over the simple SVR model, but fails to perform as well as CCNF, and when more information is encoded into the feature vector, it appears to lose any advantage over SVR at all. For these reasons, the further experiments were performed using the SVR and CCNF models only, as the comparison of the two models leads to more interesting conclusions.

Tables 5.6 and 5.7 show the results achieved using the CCNF and SVR models with both the basic and the relative feature vectors on the reduced MoodSwings dataset. We can see that overall the results on the original MoodSwings dataset are better than those on the updated dataset (either due to the larger dataset or due to the features used, or both), but similar trends can be observed in both. Relative feature representation improves the performance of the SVR model by quite a lot, and it produces results that are on the same level as CCNF. For the arousal model, CCNF performs equally well with both feature representation techniques, but the difference becomes apparent once more with the valence axis, where the basic feature representation outperforms the relative feature representation.

Table 5.6: Results for both the SVR and the CCNF arousal models, using both the standard and the relative feature representation techniques on the MoodSwings dataset

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
SVR-B	0.206	0.188	0.610	0.014	0.217	0.189	0.224	0.048
SVR-R	**0.183**	0.153	**0.697**	**0.054**	0.212	0.176	0.274	0.014
CCNF-B	0.187	0.154	0.690	0.032	**0.204**	**0.167**	**0.330**	**0.054**
CCNF-R	0.184	**0.152**	0.691	0.052	0.210	0.172	0.310	-0.018

Results on a much bigger MediaEval 2014 dataset tell a similar story (Table 5.7). We see a clear advantage of relative feature representation for the SVR model for both axes and CCNF for the arousal axis, but not for the valence axis. It is also interesting to see that while SVR with the relative feature representation was performing at the same level as CCNF on the smaller dataset, on the larger dataset it is clearly the best performing model of the four. Another interesting point is the short correlation result—with the MediaEval 2014 dataset, the short correlation for arousal is much larger than could have been expected based on the results we have seen so far, while the models for valence axis conform to the trend we have seen before.

Table 5.7: Results for both the SVR and the CCNF arousal models, using both the standard and the relative feature representation techniques on the MediaEval 2013 dataset (or 2014 development set)

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
SVR-B	0.198	0.176	0.553	0.160	0.210	0.184	0.264	0.054
SVR-R	**0.180**	**0.147**	**0.633**	0.189	**0.190**	**0.158**	**0.394**	0.043
CCNF-B	0.211	0.173	0.547	0.107	0.203	0.167	0.321	0.047
CCNF-R	0.185	0.152	0.611	**0.206**	0.210	0.172	0.311	**0.066**

Other feature representations

While relative feature representation is the best-performing feature representation in terms of short and long RMSE and long correlation; even combining it with CCNF did not seem to improve the short correlation results. As we have seen a large improvement in short correlation with the moving relative feature representation and with time delay window, it was interesting to see if CCNF would allow for the same improvement. Table 5.8 shows the results comparing SVR and CCNF with some of the better performing feature representations (which are described in Chapter 4). We can see that overall the results for short and long RMSE and long correlation achieved with either of these techniques do not outperform the relative feature representation. Nonetheless, it is clear that CCNF provides a large advantage over SVR when using these techniques, especially as measured by the short metrics. It is also evident that CCNF maintains the improvement in the results for short correlation.

It is also interesting to note that we can see a large improvement when using

the time-delay feature representation technique. While CCNF encodes some of that temporal information lost when using a bag-of-frames approach, it can still benefit from seeing the actual features from the past samples.

Table 5.8: Results comparing SVR and CCNF using several different feature representation techniques, on the updated MoodSwings dataset, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
Moving relative representation with 5s window								
SVR	0.186	0.165	0.682	0.199	0.212	0.180	0.266	0.144
CCNF	0.187	0.155	0.695	0.245	0.205	0.167	0.322	0.126
Delay samples with 4s window								
SVR	0.188	0.168	0.675	0.206	0.213	0.183	0.274	0.176
CCNF	0.186	0.153	0.693	0.287	0.208	0.169	0.320	0.229

5.4.4. Model-level fusion

Model-level fusion does not appear to have a strong positive effect, and it varies greatly depending on which machine learning technique is used and which axis we are working with. For valence, the hierarchical SVR model using relative feature representation seems to achieve one of the best results (Table 5.9), with only the CCNF model with basic representation and without the MFCC features performing better (Table 5.4). For arousal, CCNF with basic representation seems to be performing the best. Again, the results are similar or slightly worse than those achieved by CCNF with shorter feature vectors (Table 5.4).

The Euclidean distance when using SVR went down from 0.129 to 0.123 when using the basic feature representation as compared to the relative representation, but went up from 0.118 to 0.125 when using CCNF. The combination of the standard feature representation and the CCNF model was therefore the most effective solution, at least based on the Euclidean distance metric.

Table 5.9: Results comparing the CCNF approach to the SVR with RBF kernels using model-level fusion, basic (B) and relative (R) representation on the original MoodSwings dataset.

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
SVR-B	0.205	0.182	0.632	-0.003	0.216	0.182	0.207	0.058
SVR-R	0.186	0.159	0.714	-0.001	**0.204**	**0.167**	**0.304**	0.032
CCNF-B	**0.168**	**0.146**	**0.737**	**0.043**	0.209	0.171	0.263	**0.073**
CCNF-R	0.172	0.148	0.722	-0.014	0.226	0.183	0.183	0.001

5.5. MediaEval2014

Research tasks or challenges are a good way of comparing your approach to a problem to that of others. Such an opportunity arose with the MediaEval 2014 Emotion in Music task (see Section 2.3 for an introduction). As the setting of the

task is nearly identical to the approach I am taking towards the problem of emotion recognition in music, I used this opportunity to compare the relative feature representation and the CCNF with other approaches in the field. The dataset used in this challenge is described in the Section 2.4.3.

Table 5.10: Results for both the SVR and the CCNF arousal models, using both the standard and the relative feature representation techniques on the MediaEval 2014 test set

	Arousal				Valence			
	Corr	range	RMS	range	Corr	range	RMS	range
Baseline	0.18	+/-0.36	0.27	+/-0.1	0.11	+/-0.34	0.19	+/-0.11
SVR-B	0.129	+/-0.32	0.146	+/-0.06	0.073	+/-0.27	0.10	+/-0.06
SVR-R	0.148	+/-0.33	0.147	+/-0.06	0.063	+/-0.59	0.10	+/-0.06
CCNF-B	0.116	+/-0.63	0.139	+/-0.07	0.074	+/-0.29	0.10	+/-0.06
CCNF-R	0.181	+/-0.60	0.118	+/-0.07	0.066	+/-0.53	0.10	+/-0.06

As the number of submissions for this task was limited, I chose to compare the performance of CCNF to that of SVR with the RBF kernel, as these two approaches were comparatively good in at least some of the experiments. I tried to maintain a similar experimental design for this challenge too—as the development and the test sets were clearly defined for this task, the overall cross-validation was omitted, but the steps taken on each fold were reproduced here. A model was trained for each axis, using 2-fold cross-validation to pick the best parameters for training, making the results at least somewhat comparable to the experiments done with the other datasets. For the baseline method, the organizers used multilinear regression on a small feature vector consisting of only 5 features: spectral flux, harmonic change detection, loudness, roughness and zero crossing rate.

There are several interesting trends visible from the results (see Table 5.10). First of all, CCNF combined with the relative feature representation clearly outperforms all the other methods for the arousal axis, as well as the baseline method. Secondly, the spread of correlation for the CCNF model is twice as big as the one for SVR, while there is little difference between the spread of RMSE for the different methods. In fact, there is little difference in performance between the different methods and the different representations used for the valence axis.

We see the exact same ranking of the four methods in Table 5.7, which depicts the results achieved by the same methods on the development set. For these experiments I used the same experimental method as elsewhere in the dissertation, so that a direct comparison can be drawn with the rest of the work described here. The metric that resembles the correlation metric used in the Emotion in Music task best is the short correlation (s-Corr), which produces the same ranking for arousal, while the ranking for valence would be different. For valence we can clearly see that the change of feature representation results in better performance for both machine learning methods and that is evident from all the evaluation metrics used. The ranking of the two machine learning methods is less clear, and CCNF with the basic feature representation no longer outperforms SVR. The results for valence for the different methods are closer together, and apart from the short correlation metric, the top performing method seems to be SVR with relative feature repres-

entation, which is similar to the conclusion drawn from the MoodSwings dataset.

It is interesting to compare the results achieved with this dataset to those achieved with the MoodSwings dataset. This shows how much of an impact the dataset has on the performance and even the ranking of different methods. In the experiments described in the previous sections, CCNF clearly outperformed SVR with the standard feature representation, while the results with the relative feature representation were comparable between the two models. With this dataset, a very different conclusion would have to be drawn—with the standard representation the results are comparable between the two models, with a small advantage for SVR, while there was a clear difference between the two when using the relative feature representation for the arousal axis, with CCNF clearly outperforming SVR. Despite the clearly better correlation coefficient of the CCNF model with the relative feature representation, both SVR models featured a smaller deviation between different songs—half as small as that for both CCNF models. Another interesting insight is how different the ranking of the five methods would be if the RMSE was used as the primary metric, as opposed to the correlation, which was chosen for this task. It is also interesting to note that the same can be seen in the results on both the test set and the development set.

5.6. Discussion

The results achieved with CCNF are encouraging—CCNF was the best performing model in most of the experiments described in this chapter. With the MoodSwings datasets and the basic feature representation, it consistently outperformed both the standard baseline used in the field (SVR) and the more advanced CCRF model. With the relative feature representation, the results are more mixed—CCNF definitely outperforms CCRF, but the performance is often matched by SVR with the RBF kernel. Results with the bigger MediaEval dataset are also more mixed: the ranking between SVR and CCNF is less clear, but the improvement brought by the relative feature representation is substantial and the CCNF with this representation still achieves the best results.

Schmidt and Kim [2010b] used the same original MoodSwings dataset for their experiments that were based on a similar experimental design. They reported mean Euclidean distance of 0.160-0.169 which is within the same order of magnitude as the best average Euclidean distance of 0.116 achieved in these experiments. Unfortunately, even though the same dataset is used, the experimental design is slightly different and concrete comparisons are difficult to make.

5.6.1. Other insights

It appears that the models described in this chapter are reaching some sort of ceiling when it comes to the reduced MoodSwings dataset. While with the larger MediaEval dataset the results are more varied, the MoodSwings results are starting to stabilise as the models get more complicated.

The model-level fusion did not give the improvement I expected. It seems that the improvement is larger when a simpler machine learning technique is used.

This leads us to the hypothesis that there is a balance between the complexity of a machine learning method used, and the complexity of the feature vectors used. It seems that when using simpler machine learning techniques, the results can be greatly improved by spending more time carefully designing the features employed. On the other hand, when a more advanced method is used we get diminishing returns from more complex feature vectors. It would therefore appear that the relationships I uncovered by building complex features can also be implicitly learned with more advanced techniques.

Both CCRF and CCNF can be compared with the time window feature representation described in Section 4.3—both the two machine learning models and the feature representation technique are trying to expose the features of the surrounding samples to the decision process for the current sample. While CCNF with the basic feature representation performs better than SVR with delay window feature vector as measure with most evaluation metrics, short correlation of the latter model is still higher. When CCNF is combined with delay window feature vector, we see a large increase in short correlation, which implies that a modification to CCNF where the input of surrounding samples is available to the neural layers of the current sample might improve the results even more.

The experiments with smaller feature vectors have shown that the size of the feature vector plays a major role in the performance of CCNF. The fact that better results were achieved by omitting a whole class of features shows that there potentially is some redundancy between the four sets of features used in the original MoodSwings dataset. It would, therefore, be advisable to investigate some feature selection or sparsity enforcing techniques.

5.7. Conclusion

In this chapter I focused on the machine learning algorithm's element of continuous dimensional emotion recognition in music. I described three different models: SVR, CCRF and CCNF. SVR is the most commonly used machine learning technique in the field, while CCRF and CCNF have never before been used for the task of emotion recognition in music. The latter two algorithms attempt to re-encode some of the temporal information lost when using bag-of-frames approach—there is a clear temporal aspect in musical emotion, and incorporating it unsurprisingly improved the results. All three models are publicly available to download and use with publicly available datasets, making it easy to use as comparison for other researchers.

First of all, I showed the importance of correct choice of training parameters and kernel for SVR, justifying my choice of RBF kernel throughout all the experiments. I also compared SVR, CCRF and CCNF using a variety of datasets and several feature vector representation techniques.

Using the original MoodSwings dataset and basic feature representation, CCNF outperforms the other two models (reducing long RMSE by 14.4%, short RMSE by 19.7% and increasing long correlation by 14.6% when compared to SVR). With relative feature representation, both SVR and CCNF achieve comparable results.

A similar effect is also observed with the reduced MoodSwings dataset with OpenSMILE extracted features. When the MediaEval 2013 dataset is used, the results are more mixed, but CCNF combined with relative feature representation is again the best performing model on the MediaEval 2014 test set.

I also tested SVR and CCNF with two of the other best performing feature representations described in Chapter 4—moving relative representation (5 s window) and delay (4 s) window. Both of these were tested with the reduced MoodSwings dataset and resulted in CCNF outperforming SVR, but they both also maintained the same feature that they achieved in the original experiments—high short correlation, in this case, up to 0.287 for arousal (39.3% improvement over SVR) and 0.229 for valence (30.1% improvement over SVR).

Multi modality

6

The easiest, but by no means easy, approach to automatic emotion recognition in music is to take the acoustic data as it is, extract a set of fairly low level features from it and then apply a reasonable machine learning method to extract the underlying patterns from the data. While such a solution can achieve good results, especially for the arousal axis, it is without doubt a rather simplified view of songs and our experience of them. It therefore leaves plenty of space for improvement. This chapter describes some work that I did with an attempt to separate out the different modalities (background music, singing voice and lyrics) present in a song and look at them on their own before combining everything into a single, multi-modal system. Unfortunately some of the underlying technologies that are required for a more in-depth multi-modal analysis remain poorly understood, so this chapter explores the issues and demonstrates a “proof of concept" rather than offering a complete solution.

Some of the work described in this chapter is under review for publication.

6.1. Separation of vocals and music

There exists some evidence that musical and vocal emotion are processed by different parts of the brain and are perceived differently [Peretz, 2010]. This suggests that the vocals and the background music might also have a different effect on the expressed emotion of a song. In this section I describe a system that attempts to separate these two modalities (vocals and background music), analyse them separately and then use the features extracted from both to train a machine learning model for emotion recognition in sung music.

6.1.1. Separation methods

Two main audio separation techniques were used in my work—REpeating Pattern Extraction Technique (REPET) by Rafii and Pardo [2013] and "Voiced+Unvoiced"-Instantaneous Mixture Models (VUIMM) by Durrieu et al. [2011]. Both of these methods are aimed at the separation of voice (or the main melody line) and the background music, but they differ in the approach that is used. While the assumptions underlying both of these methods are somewhat simplifying and inevitably

break down in some of the songs, these separation techniques are still state-of-the-art enough to provide me with a good basis for my experiments.

REPET

REPET is based on the idea that a song is generally composed of two main parts–a repeating background signal and a varying foreground signal (a singer with a repeating accompaniment or speech with a repeating background noise). This leads to a method that is simple, fast and does not require any external information. It contains three main steps: identify repeating patterns, derive repeating models and extract repeating patterns. There are several modifications of the main technique that I have used in my work: adaptive REPET [Liutkus et al., 2012], REPET with segmentation [Rafii and Pardo, 2013], and REPET-SIM [Rafii and Pardo, 2012]. All of them work on the spectrogram of a signal and follow the same main structure of the algorithm.

The original technique by Rafii and Pardo [2013] has three stages. In the first stage, the signal is transformed into a spectrogram and the beat spectrum is extracted by estimating the beat spectrum bands and finding the repeating period p. In the second stage the spectrogram is segmented into segments of length p. The repeating element S (of the same dimensions) is constructed by taking the element-wise median (for each element in the matrix) of all the segments in the spectrogram. In the third stage the repeating spectrogram W is derived by taking the element-wise minimum between the repeating segment S and each original segment of the spectrogram (the min function is used to avoid negative spectral values when the non-repeating spectrogram is extracted). Finally, a soft time-frequency mask M is derived from the repeating spectrogram W by element-wise normalisation of W by the original spectrogram so that the repeating elements would have values close to 1 and the non-repeating values close to 0. The mask is then applied to the short-time Fourier transform of the original mixture, inverted into the time-domain and the non-repeating signal is obtained by subtracting the repeating signal from the original signal in the time-domain. To cope with varying repeating structures the algorithm is easily extended by applying it to individual segments of the signal.

For a more elaborate and reliable algorithm that can cope with varying repeating structures (e.g. verse versus chorus) Adaptive REPET is introduced by Liutkus et al. [2012]. In this method, the background is assumed to be locally-spectrally periodic, with a time-varying period. Instead of the beat spectrum that is extracted in the first stage of REPET, a beat spectrogram is extracted in Adaptive REPET. The beat spectrogram is calculated by taking an average of all the spectrograms of all the frequency channels of the original signal, and is then used to estimate a time-varying period $T_0(t)$ for each time-frame. In the second stage, for each time-frame t_i a median of all the frames at the period $T_0(t_i)$ is taken. This way, instead of a repeating segment, we get a repeating spectrogram. This repeating spectrogram is then further refined in stage 3 where the element-wise minimum is taken between the original spectrogram and the repeating spectrogram (instead of the repeating segment). The remaining steps are exactly the same as in the original algorithm.

Finally, REPET-SIM was introduced by Rafii and Pardo [2012] so that it could cope

with non-periodically repeating structures. The REPET-SIM algorithm is actually quite similar to the Adaptive REPET algorithm described above. In the first stage, instead of the beat spectrogram, a similarity matrix is extracted from the original spectrogram—we are now looking for repetition of the frames in the original signal. Not all the repetitions of a frame are of interest to us—a maximum number of repetitions is defined, as well as the minimum threshold for similarity and the minimum allowed distance between two consecutive repeating frames. The idea is that the non-repeating "voice" frames are relatively sparse and varied, therefore by identifying the repeating elements we will identify the repeating background. In the second stage, instead of taking the median of all the frames at a particular period, we are taking a median of all the similar frames, thus, again, creating a repeating spectrogram. The third stage is identical to the third stage of the Adaptive REPET algorithm.

VUIMM

The other method that I have used for separating the vocal part from the background music is VUIMM, developed by Durrieu et al. [2011]. Unlike the REPET family, VUIMM works with a mid-level representation—instead of focusing on repeating and non-repeating parts of the spectrum, it assumes that the vocal part in an audio is the pitched signal, while the residual is the background music. The algorithm works by estimating pitch, which is a mid-level representation, as opposed to a spectrogram, which is a low-level representation.

The VUIMM is built upon a general Instantaneous Mixture Model (IMM). In this model, the audio signal is assumed to be an instantaneous mixture of different contributions—in particular, a signal of interest and a residual. The two signals are considered independent from each other, and so the power spectrum of the whole signal can be considered as simply the sum of the power spectra of the two signals. The power spectrum of each time frame in each frequency bin can further be decomposed into the excitation spectrum (the source) and a spectral shaping envelope (a filter). The filter part makes it possible to adapt to and discard various secondary effects, such as recording conditions, tempo, intonations for a voice, etc. The source can be further defined to be a combination of the spectral envelope (timbre properties) and the pitch content. The IMM enables us to separate the source from the filter and then to extract the pitch information from the source.

As IMM is particularly suitable for signals that consist of one main harmonic instrument (for example singing voice) backed by an accompaniment, the general model can be adapted to model exactly that. The source or the voice part of the signal is then modeled by the framework described above, while the background music is modeled by a non-negative matrix factorization. The model as it is described now would not be able to capture the aperiodic or the unvoiced components of the source signal. To fix that, the matrix representing the spectral envelope is modified to include the spectral envelope for the white noise which is only added once all the other parameters are estimated, thus avoiding capturing the noise elements of other musical instruments present in the signal.

The VUIMM is an iterative algorithm, as it needs to estimate the non-fixed para-

meters of the model (i.e. not the spectral envelope matrix or the matrix of smooth filters). While the optimal number of iterations is song-specific, in all the experiments described in this thesis, the number of iterations was kept at 100, as suggested by Durrieu et al. [2011], as this seemed to have been the point at which the performance of the algorithm plateaus.

For more details on the model, see Durrieu et al. [2011].

6.1.2. Methodology

Based on the insights from Chapter 5, in this section I used two machine learning models: Support Vector Regression (SVR) and Continuous Conditional Neural Fields (CCNF), as I believe their comparison provides the most interesting results.

The experimental design to compare the effects of music-voice separation was kept identical to the experimental design used in the rest of the dissertation (see Sections 4.2 and 5.4.1). A 5-fold cross validation was used for training-testing, keeping the folds consistent throughout the experiments described both in this chapter and in the rest of work. Then another 2-fold cross-validation was used within each training set to determine the different parameters required for training. The final results are the average results achieved by each fold, in each separate cross-validation set.

Three different datasets were extracted for each separation technique. First of all, the voice-music separation was done on a set of songs, producing two WAV files for each input song with the vocal track in one and the background music in the other. Then the standard set of features (see Section 4.1) was extracted with the OpenSMILE tool separately from the two resulting sets of audio files, thus producing two datasets. The final dataset was produced by combining the features extracted from the voice and the background track into the same vector, in other words by performing feature fusion. Unless it is explicitly specified, whenever one of the separation methods is mentioned, the feature fusion technique was used to create it, i.e. the dataset contains features from both sources.

6.1.3. Results

The analysis of the results achieved using feature vectors built from the separated audio are described in the sections below. As elsewhere in this dissertation, I am using short and long correlation and RMSE to report the results. Long RMSE and correlation are provided as a reference to other work in the field, while short RMSE and short non-squared correlation is provided to give a better insight to the performance of the models.

Separate feature vectors

The first experiment that I did with the separated audio files was to use the single-modality feature vectors on their own to build a machine learning model. Tables 6.1 and 6.2 show the performance of such models using the two machine learning techniques. While no signal separation method is consistently better than the other

ones, and the ranking differs with each evaluation metric, separation of music and vocals still produces several interesting observable trends in the results.

Music-only analysis, or the removal of the vocals, seems to improve the performance of the SVR model (Table 6.1), at least with the basic feature representation, when compared to the original signal. The vocals-based feature vector, on the other hand, leads to inferior performance for both axes. Results for the valence axis are worse overall, and especially bad with the vocals-based basic feature representation, while the relative feature representation seems to ameliorate the effect. Relative feature representation seems to dampen the positive and negative effects of signal separation and the results are similar to those achieved with the analysis of the original signal.

Table 6.1: Results for the different music-voice separation techniques using the basic and relative feature representations with the single modality vectors, SVR with RBF kernel, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
Basic feature vector - music								
REPET (seg)	0.192	0.169	0.671	-0.012	0.217	0.186	0.224	0.036
Ada-REPET	0.191	0.169	0.666	-0.008	0.216	0.187	0.225	0.059
REPET-SIM	0.188	0.166	0.679	-0.022	0.219	0.188	0.217	0.037
VUIMM	0.204	0.185	0.619	0.011	0.218	0.189	0.213	0.050
Basic feature vector - vocals								
REPET (seg)	0.235	0.218	0.495	0.010	0.230	0.203	0.133	0.050
Ada-REPET	0.230	0.212	0.512	-0.010	0.228	0.201	0.145	0.051
REPET-SIM	0.237	0.220	0.484	0.006	0.234	0.203	0.115	0.047
VUIMM	0.240	0.220	0.472	0.017	0.235	0.206	0.100	0.031
Relative feature vector - music								
REPET (seg)	0.182	0.151	0.701	0.042	0.223	0.186	0.204	0.056
Ada-REPET	0.185	0.153	0.694	0.042	0.215	0.180	0.240	0.020
REPET-SIM	0.175	0.149	0.727	0.034	0.217	0.181	0.236	0.041
VUIMM	0.187	0.158	0.687	0.035	0.213	0.176	0.246	0.006
Relative feature vector - vocals								
REPET (seg)	0.186	0.156	0.694	0.018	0.225	0.189	0.211	0.009
Ada-REPET	0.185	0.153	0.691	0.008	0.225	0.185	0.197	0.022
REPET-SIM	0.190	0.157	0.673	-0.009	0.219	0.182	0.224	0.018
VUIMM	0.209	0.176	0.608	0.036	0.226	0.189	0.208	-0.009

A similar effect can be seen with the CCNF models (Table 6.2), although to a smaller extent. Arousal models based on music analysis and basic feature representation achieve lower short RMSE, and possibly slightly larger long correlation, while there is no effect measurable with the other two metrics. Relative feature vector based arousal models for both modalities achieve the same performance as simple CCNF model using the original signal, while all the models for valence perform worse than the original CCNF model.

Table 6.2: Results for the different music-voice separation techniques using the basic and relative feature representations with the single modality vectors, CCNF, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
Basic feature vector - music								
REPET (seg)	0.185	0.153	0.697	0.125	0.217	0.182	0.248	0.058
Ada-REPET	0.186	0.155	0.673	0.028	0.210	0.173	0.295	0.005
REPET-SIM	0.185	0.146	0.697	0.047	0.217	0.175	0.248	-0.006
VUIMM	0.182	0.151	0.700	0.063	0.214	0.179	0.277	0.083
Basic feature vector - vocals								
REPET (seg)	0.185	0.152	0.688	0.043	0.211	0.177	0.297	0.035
Ada-REPET	0.187	0.153	0.688	0.080	0.219	0.180	0.251	0.054
REPET-SIM	0.185	0.156	0.688	0.079	0.211	0.169	0.297	0.042
VUIMM	0.203	0.167	0.656	0.049	0.226	0.188	0.180	0.060
Relative feature vector - music								
REPET (seg)	0.187	0.152	0.689	0.064	0.229	0.188	0.219	0.050
Ada-REPET	0.185	0.155	0.696	0.057	0.215	0.178	0.264	-0.005
REPET-SIM	0.187	0.149	0.689	0.017	0.229	0.185	0.219	0.038
VUIMM	0.184	0.150	0.700	0.057	0.221	0.182	0.223	-0.065
Relative feature vector - vocals								
REPET (seg)	0.183	0.151	0.690	0.020	0.219	0.181	0.249	-0.009
Ada-REPET	0.188	0.153	0.673	0.096	0.228	0.191	0.219	0.041
REPET-SIM	0.183	0.157	0.690	0.044	0.219	0.183	0.249	0.024
VUIMM	0.210	0.173	0.620	0.085	0.225	0.188	0.202	0.014

Feature-level fusion

Table 6.3 shows the results of the combined, feature fusion vector containing both the features extracted from the vocals and from the background music and trained with SVR. While with the standard feature representation it is difficult to pinpoint the best technique, especially for the valence axis, one thing is clear—performing separation of vocals and the background music improved the results when compared to the same analysis and training done on the original audio signal. It is also interesting to note the effect on valence models—while with single-modality vectors, results of valence models deteriorated, with the combined feature vector they are now either as good as the original analysis or better. A similar effect can be seen with the relative feature representation, except that for the arousal axis there is a stronger preference for the VUIMM separation technique.

When the same approach is tested with the CCNF model (see Table 6.4), we see a similar effect on the arousal axis—voice and music separation gives a clear advantage over simple audio analysis with the basic feature vector. Results for the valence axis are much less convincing—none of the separation techniques seem to give a clear advantage over the simple audio analysis. The results using the relative feature representation tell a similar story—for the arousal axis, we can see that VUIMM offers a convincing improvement over the basic audio feature ana-

Table 6.3: Results for the different music-voice separation techniques using the basic and relative feature representations with the feature-fusion vector, SVR with RBF kernel, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
Basic feature vector								
Original	0.206	0.188	0.610	0.014	0.217	0.189	0.224	0.048
REPET (seg)	**0.189**	**0.168**	**0.676**	0.010	0.219	0.188	0.215	**0.071**
Ada-REPET	0.191	0.170	0.665	0.011	0.217	0.187	0.227	0.054
REPET-SIM	**0.189**	0.169	0.674	-0.003	**0.213**	**0.185**	**0.246**	0.040
VUIMM	0.196	0.177	0.649	**0.029**	**0.213**	**0.185**	0.235	0.043
Relative feature vector								
Original	0.183	0.153	0.697	0.054	0.212	0.176	**0.274**	0.014
REPET (seg)	0.187	0.157	0.684	0.038	0.216	0.180	0.249	0.037
Ada-REPET	0.182	0.151	0.700	0.029	0.211	0.175	0.266	0.009
REPET-SIM	0.182	0.152	0.696	0.017	0.212	0.175	0.273	**0.044**
VUIMM	**0.178**	**0.149**	**0.717**	**0.055**	**0.210**	**0.173**	0.271	0.020

lysis, which also relates to the results achieved using SVR. For the valence axis, the message is a lot more mixed, and it is not clear if separation of vocals and music provide any benefit. Just as with SVR models, no information seems to be lost during the separation process, and the performance of valence models is as good or slightly better than the original analysis.

Table 6.4: Results for the different music-voice separation techniques using the basic and relative feature representations with the feature-fusion vector, CCNF, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
Basic feature vector								
Original	0.187	0.167	0.690	0.032	**0.204**	**0.167**	0.330	0.054
REPET (seg)	0.178	0.147	0.714	**0.095**	0.210	0.175	0.285	**0.110**
Ada-REPET	0.178	0.149	0.717	0.018	**0.204**	0.169	**0.332**	0.055
REPET-SIM	**0.172**	**0.145**	**0.732**	0.048	0.207	0.169	0.310	0.061
VUIMM	0.178	0.147	0.712	0.082	0.209	0.172	0.296	0.057
Relative feature vector								
Original	0.184	0.152	0.691	0.052	**0.210**	**0.172**	**0.310**	-0.018
REPET (seg)	0.183	0.153	0.699	0.062	**0.210**	**0.172**	0.293	-0.030
Ada-REPET	0.185	0.152	0.694	0.061	0.218	0.180	0.268	-0.024
REPET-SIM	**0.173**	**0.145**	0.724	0.003	0.213	**0.173**	0.288	-0.017
VUIMM	0.177	**0.146**	**0.735**	**0.102**	**0.211**	0.176	0.293	**0.060**

Overall, we see the same trends with the two machine learning solutions and the two feature representations as we saw in Chapter 5: when using SVR, we can see a clear advantage of using the relative feature representation, and we can clearly see an advantage of using CCNF compared to SVR, but the combination of CCNF and relative feature representation does not seem to offer any improvement over

basic feature representation.

6.1.4. Conclusions

In this section I have described four different music and voice separation techniques: REPET (segmented), Adaptive-REPET, REPET-SIM and VUIMM. I have showed how each of the two modalities (music and voice) can be used on their own to train a machine learning model—arousal models derived from acoustic analysis of music only (without the vocals), achieve results that are better than simple acoustic analysis of the original signal, reducing long RMSE by up to 8.7%, short RMSE by up to 12.2% and increasing long correlation by up to 14.8% (using REPET-SIM separation, basic feature representation and SVR model). Valence models, on the other hand, suffer from such a single-modality analysis.

Extracting acoustic features from the two modalities separately and then combining them into a single feature vector improves arousal results further for both machine learning models and using both the basic and relative feature representations. Basic feature representation models using SVR achieve similar results as above, while other SVR models are improved by around 2%. CCNF models are improved by more—for basic feature vectors, long RMSE is decreased by 8.0%, short RMSE by 13.2%, and long correlation is increased by 6.1%; for relative feature vectors, long RMSE is decreased by 6.0%, short RMSE by 4.6% and long correlation is increased by 6.0%. Valence models are improved slightly when using SVR, reducing both short and long RMSE by up to 2% and increasing long correlation by up to 9.8%, while CCNF-trained models perform either worse than or as well as the models trained on simple acoustic features.

6.2. Lyrics

While models based on audio analysis can achieve sufficient performance for the arousal axis, the results for the valence axis leave a lot to be desired. A similar situation can be seen in a related field—sentiment analysis in text—except here valence is much easier to predict than arousal. It suggests that there is untapped potential that can be reached through the analysis of lyrics. There have been attempts at exploring this idea for the overall emotion prediction in music with positive results (see Section 2.2.4). Unfortunately, it is a lot more difficult to reuse the techniques for the analysis of lyrics for the overall emotion prediction than it is to reuse those used for audio analysis. Further sections will explain why that is the case, and my suggested approaches to music emotion prediction based on lyrics.

The following techniques, when used on their own, would obviously only work for songs that contain vocals. Section 6.3 describes how the acoustic features can be combined with the analysis of lyrics to reach a multi-modal solution that exploits the information extracted from lyrics when it is present, but does not fail when no such information exists.

6.2.1. Techniques

In order to investigate the potential of the analysis of lyrics for emotion tracking in music, I experimented with a variety of techniques, both machine learning based and not.

As a baseline, I looked into the performance that can be achieved by a simple mapping of the words occurring in a song to their annotations on the 3-dimensional emotion space. This is achieved by using an affective norms dictionary and such features are commonly used when analyzing lyrics of the whole song [Neumayer and Rauber, 2007; Hu and Downie, 2010a,b; Chuang and Wu, 2004; Yang and Lee, 2009; Hu et al., 2009b]. The basic approach is to simply average the ratings for a particular axis for all the words that are sung in a particular time period (in this case, a second). This is a naive way of looking at the effect of lyrics. It is clear that we do not simply consider each word and its emotional content on its own and constantly reevaluate our judgment based on new and expected information. A somewhat more grounded approach is to take an exponential average of all the words that have been sung so far—in this approach, the "current" words still carry the most weight, but all the previously sung words have influence over the emotion label in question.

$$\text{label}_T = \alpha * \text{average}_T + (1 - \alpha) * \text{label}_{T-1}$$

While the above-mentioned approaches can be expected to work reasonably well for songs that have a near-continuous singing line, it would obviously be insufficient in sections that only contain the background music without a singing voice. Exponential averaging of emotional labels ameliorates this problem somewhat, as the effect of words sung in the past would still have an effect, even if that effect would be diminishing with time. This method would still fail if the musical sections were long and would have no effect at the intro of a song. Another solution to this problem relies on a similar idea to that of a relative feature representation used for the acoustic features in machine learning solutions—the average sentiment expressed by the lyrics of a song can also be incorporated when the emotion label for a particular sample is computed. A weighted average between the song average and the average of annotations for a particular second could be used when a singing line is present, and simply the average of the song labels could be used for sections without a singing line.

Finally, a machine learning solution can be used to uncover the hidden relationships between the emotion labels as well as to enable us to use more information to reach a good final emotion label. All of the feature vector building techniques described in Chapter 4 are still applicable when building a lyrics analyzer, some of them replacing or mimicking the motivation behind the techniques described above.

LDA

While some of the most popular Information Retrieval (IR) techniques (see Section 6.2.1 below) use large numbers of dimensions to represent documents, there is a

family of algorithms that attempt to model latent topics in a corpus of documents. Such semantic representation of terms is a popular technique used not just for analysis of text in general, but also for sentiment analysis in lyrics [Xia et al., 2008; Laurier et al., 2008; Logan et al., 2004; Yang et al., 2008].

Latent Dirichlet Allocation (LDA) introduced by Blei et al. [2003] is one such algorithm. It is based on an unsupervised machine learning algorithm that attempts to identify hidden topic information in large document collections. It does not require prior knowledge of the number of topics in the corpus.

Each document $\mathbf{w}_j = (w_1, \ldots, w_N)$ consisting of N words in the collection $D = \{\mathbf{w}_1, \ldots, \mathbf{w}_M\}$ is represented by a probability vector θ_j which denotes the topic distribution for document j. Each topic is described by β_t—a multinomial distribution vector representing V-dimensional probability (where V is the number of unique words in the corpus) with $\beta_{t,v}$ standing for the probability of generating word v given topic t. α is used as a T-dimensional positive parameter vector of the Dirichlet distribution over θ_j.

LDA assumes the following generative process for each document $\mathbf{w}$ in a corpus D:

1. Choose $N \sim \text{Poisson}(\xi)$
2. Choose $\theta \sim \text{Dir}(\alpha)$
3. For each of the N words w_n:
   (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$
   (b) Choose a word w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n

The dimensionality k of the Dirichlet distribution, or the number of topics is assumed to known and fixed, and the Dirichlet random variable θ is defined to be a $(k-1)$-simplex, or, in other words, all of its elements have to be non-negative and sum to 1. The joint distribution of a topic mixture model θ can be derived as:

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta) \tag{6.1}$$

Integrating over θ and summing over z we obtain the marginal distribution of a document:

$$p(\mathbf{w}|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta) \right) d\theta \tag{6.2}$$

Parameters α, β are corpus level parameters and are sampled once, θ_d are document-level variables sampled once per document and variables z_{dn}, w_{dn} are word-level variables sampled once for each word in a document. This therefore gives LDA a three-level representation. Such hierarchical models are often referred to as *conditionally independent hierarchical models* or *parametric empirical Bayes models* and so

empirical Bayes approaches can be used for estimating the parameters used in the model.

The key inferential problem for LDA is to compute the posterior distribution of the hidden variables:

$$p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)}{p(\mathbf{w}|\alpha, \beta)} \tag{6.3}$$

Unfortunately, the denominator of Equation 6.3 is intractable due to the coupling between θ and β. A variety of approximation algorithms exist that can be used to approximate the inference. Variational inference approximation $O(N^2k)$ is used in the original algorithm giving a lower bound for $p(\mathbf{w}|\alpha, \beta)$, which can then be used to find parameters α, β that maximise the log likelihood of the data.

LDA gives two vectors of interest for us—the topic distribution for each document and the topic distribution for each word. These two can then be used in the same way as affective norms for words, described above.

The original algorithm is a problem of Bayesian inference and it iterates through the whole corpus several times until it forms the solution. There are other approaches that use Gibbs sampling (see Griffiths and Steyvers [2004]), expectation propagation (see Minka and Lafferty [2002]) or even single-pass approaches where only one pass over the whole corpus is made, but each document is analysed several times before moving on to another document (see Sato et al. [2010]). In this work I used a C implementation of the original algorithm[1].

Affective norms dictionaries

There are a number of different affective norms dictionaries available for researchers to use (see Section 2.1.4). As I am interested in both the arousal and valence axes, the selection is limited, as a lot of sentiment analysis in text is focused on the valence axis only. For my work on the analysis of lyrics I chose to use two affective norms dictionaries. ANEW is a good choice as it is widely used in the field and so can function well as a baseline method. Unfortunately, it is quite small (it contains only just over 1000 words) and so its use can be limited. I therefore decided to use the dictionary collected by Warriner et al. [2013] as it is based on the same methodology, but contains nearly 14,000 words annotated on all three axes: arousal, valence and dominance.

Other techniques

There are some other IR techniques that frequently get used in sentiment analysis, but which I chose not to use in my work, for reasons that I will explain below.

One of the most widely used techniques is Vector Space Models (VSM). The idea here is to represent a body of text in a multi-dimensional space. It is a bag-of-words (BOW) approach, where all the words in a document are taken individually, and their ordering is ignored. In most cases, each word (or a term) is considered

[1] http://www.cs.princeton.edu/~blei/lda-c/

as a separate axis, all dimensions considered perpendicular to each other. Several different term weighting techniques exist to position the vectors in the space (Boolean weighting, Frequency weighting, etc.), and a document or a query is then represented by the vector coefficients, which can and often do get used directly as feature vectors for machine learning solutions [Hu and Downie, 2010b; Neumayer and Rauber, 2007; McVicar et al., 2011; Wang et al., 2011; Hu et al., 2009a; Schuller et al., 2010; Guan et al., 2012; He et al., 2008]. While the idea is simple and obviously breaks when we are dealing with synonyms or words that have several meanings, the approach produces good results that are hard to beat with more complicated techniques.

Another similar technique is n-grams. Here we are interested not in individual words but in the (most common) word sequences. The most common size of an n-gram is 2 or 3, which would capture some of the negation (that would otherwise be lost in a BOW approach), some of the modifiers (e.g. “very", “little", etc.) and some of the more common shorter phrases [He et al., 2008; Guan et al., 2012; Hu and Downie, 2010a]. As the size of an n-gram increases, the size of the vocabulary grows rapidly, and while it can then capture longer phrases, we get diminishing returns as useful information gets drowned by the noise.

While both of these techniques can produce great results, they rely on large datasets for training. As the feature vector in such a system would have thousands or tens of thousands of elements, a machine learning model would be prone to overfitting (learning patterns specific to the training set and not the general, global patterns) if the training set was too small. It might also fail to extract any useful information, if not enough examples of specific patterns would be present. As you can see from Section 6.2.2, the datasets I am dealing with in my work are small, especially compared to the standard size of an IR system, which is why I decided that it would be unfeasible to use VSM or n-grams for my work. This problem would be made even worse by the fact that such vectors would be incredibly sparse when we only consider words appearing in one second of a song—analysis of short messages (e.g. Twitter) is already challenging, and this would make it even more difficult to achieve positive results.

6.2.2. Methodology

All the work described in Section 6.2 is focused on just one of the three datasets mentioned in this thesis—the MoodSwings dataset. As described in Section 2.4.1, the dataset used in this chapter will be a slightly smaller version of MoodSwings—only using the songs that I have managed to acquire. Another small modification to the dataset is the removal of songs that do not contain lyrics at all, or where the vocals only correspond to utterances that are not actual words (e.g. “la la la")—while such songs are useful for testing multi-modal systems, they are not of any use for testing systems that rely on lyrics only. The final size of the dataset is therefor 186 songs.

In the ideal scenario, a system that relies on the analysis of lyrics would have an automatic singing voice transcriber in place. As such technology is not of sufficiently high quality yet, for the purpose of this work I chose to manually transcribe

the lyrics of the songs I was using. This obviously limits the size of the dataset that I can work with and the MoodSwings dataset seemed like the obvious choice—it was reasonably small and contained popular songs, which meant that the full lyrics of the songs were also available.

Tagging

All the songs in the MoodSwings dataset were transcribed completely manually on a second-by-second basis (requiring several days of work). Each word is represented as a lemma that could be found in the Warriner Affective Norms (as well as ANEW) dictionary, with words that span several seconds or that lie on the boundary between several seconds repeated in each slot where the word appears. The mapping from the actual words to the lemmas found in the Warriner dictionary was done manually—most of the inflectional morphology was removed, i.e. plurality, tenses, degrees of comparison (comparative and superlative), etc. While various automatic stemming techniques exist and manual stemming would be of little use for an actual system, given the small size of the dataset I did not want any shortcomings in stemming and normalisation techniques to have an effect on the results.

The transcription only covers the length of the extract that is also annotated with the emotion labels. The resulting annotation is stored in a CSV file with words appearing in the same second separated by a space, 4773 words in total. The lyrics dataset also contains a text file for each song with its full lyrics, that were extracted online.

Training

The kind of processing or training required for the methods described above (Section 6.2.1) can be separated into 3 groups: no training, external training and standard training.

The basic, affective norms dictionary-based techniques require no training and can be applied to the whole dataset directly. Care must be taken with approaches like these to not overfit to the data on hand. As there are only a handful of such techniques described here, and they are used more of a baseline, the risk needs to be mentioned, but the results can still be used. When a parameter is required (e.g. for exponential averaging), a set of them is tried out and results for all different parameters used are showed in the corresponding Results section.

All the approaches that rely on LDA require an additional training process—the building of the LDA model. The implementation I use in my work (see Section 6.2.1) is an iterative learning algorithm that goes through all of the documents in each iteration updating the various parameters, and stops either after a maximum number of iterations is reached or when the results converge. In my experiments, I used random topic model initialization with 50 topics. The maximum number of iterations was set to 100 (but that number was never reached in any of the trainings), the convergence criterion was set to 0.0001, and the parameter α was estimated at each iteration together with the topic model.

An extremely important part of the algorithm is the document corpus on which the algorithm is trained, as the latent topics will be modeled on the topics present in the corpus. Two options are available to us: a general corpus that should reflect the global topic distribution; and a specialised corpus that consists of a representative set of documents of the type that we are interested in, for a more specialised set of topics. For the first option I used a small corpus of 2246 articles from Associated Press (AP) provided with the implementation of LDA. The AP dataset consists of newspaper articles used as is, without any stemming or normalisation, with a vocabulary of 10,743 words.

The second option was tested with a large corpus of 237,662 lyrics of popular songs—the musixMatch dataset[2]. This dataset is part of the Million Song Dataset project and is presented in the bag-of-words format with only the top 5000 words and their frequencies in each document. This way the dataset is still usable for most IR research and copyright issues are avoided. Before the top 5000 words are picked, all of the lyrics undergo a simple normalisation process and are stemmed, and I used the same script for normalising and stemming my dataset for the inference using LDA trained on the musixMatch dataset. It is clear from the top 5000 words that there is a large presence of songs in languages other than English (with French being the second most common language), but the stemming and normalisation is focused on English words only.

The dataset comes split into two parts—one for testing (27143 items) and one for training (210,519 items). From these I have designed two datasets for LDA training—in the first one, MXM100, I chose 100,000 random documents from the training set, and for the second one, MXM, I combined both the training and the testings sets into one dataset containing 237,662 documents. The AP dataset is used as is, with all of the 2246 documents. The trained models are then used to infer the topic distribution for both the sets of words appearing in each second individually and over the whole 15 s extract.

Finally, the standard (in this thesis) experimental design is applied to machine learning solutions described in this section. A 5-fold cross validation is used with fixed folds across the different experiments with 2-fold cross-validation inside a training set for picking any hyper-parameters required. Both CCNF and SVR with RBF kernel are used to model the affective content of the songs. The results are reported as an average across all of the folds.

6.2.3. Results

As described in the section above, the songs that contain no singing voice had been removed from the MoodSwings dataset, but, as could be expected from a standard pop-song, not all 1-second-samples contain lyrics (or singing voice). On average, 11.4 out of 15 s in each song contained vocals, while for some of the extracts that number was much lower—only 2 seconds, and on average there are 3.6 seconds of an extract without any vocals. The full distribution of the proportion of an extract containing lyrics in this dataset can be seen in Figure 6.1. Figure 6.2 shows

[2] http://labrosa.ee.columbia.edu/millionsong/musixmatch

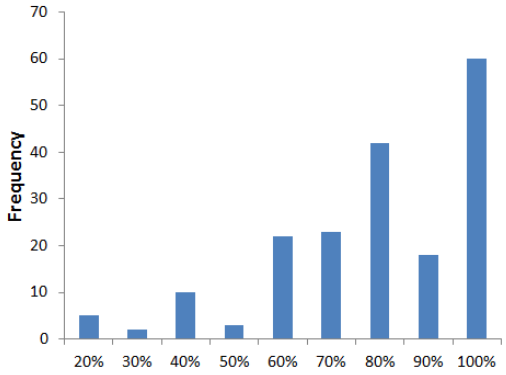


Figure 6.1: A histogram showing the proportion of extracts that contain lyrics in the reduced MoodSwings dataset.

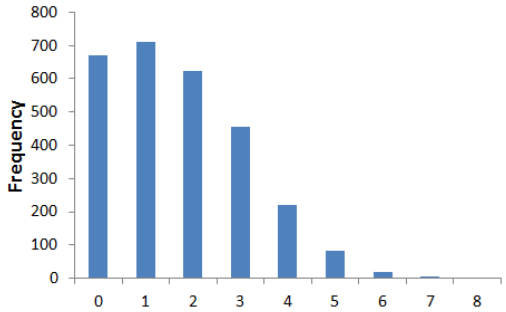


Figure 6.2: A histogram showing the distribution of the number of words present in a second of a song in the reduced MoodSwings dataset.

the distribution of the number of words appearing in a second of an extract—we can see that while most of the time we find 1-3 words in a 1-second-sample, this number can go up to 8 words per second. Both of these graphs reiterate the need for our methods to cope with the time periods that only contain the background music without any vocals—while predicting neutral emotion would probably give acceptable results, it would clearly not be the desired behaviour of a model predicting emotion in a song.

The following subsections report the results achieved with various techniques described in Section 6.2.1. The results, as elsewhere in the thesis are reported using the short and long RMSE and correlation metrics, with a larger emphasis on RMSE in this section. For non-machine learning methods the results are simply calculated over the whole dataset, while machine learning methods require cross-validation, and so the results are averaged over the folds.

Simple averaging

As described in Section 6.2.1, the simplest way to analyze lyrics is simply to average the affective values of words appearing in each second of a song. As not all the words are present in the affective norms dictionaries used in this work (e.g. no pronouns, conjunctions, etc.), the average in the following experiments is taken

only over the words found in the dictionary, and not over the total number of words appearing in a particular second. The computed average of valence values for the words is then used as the valence label for that second, and the same is done with the arousal values.

Table 6.5: Results for the simple averaging technique with both affective norms dictionaries, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
ANEW	0.352	0.316	0.003	-0.012	0.323	0.297	0.005	-0.014
Warriner	0.444	0.407	0.000	-0.004	0.320	0.301	0.018	-0.004

As can be seen from the Table 6.5 the results achieved by this simple model are less than ideal. The correlation values are around zero, but the interesting observation is that for the first time we can see much better RMSE results for the valence axis than for the arousal axis. Neither of these two points should surprise us, as lyrics analysis was supposed to improve valence results and the simplest possible technique with sparse data should not be expected to achieve great results. It is also interesting to note that despite a much lower rate at which the words are found in the ANEW dictionary (9.6% as opposed to 45.9% for Warriner dictionary), valence results for both of the dictionaries are virtually the same, and ANEW seems to be a lot better than Warriner dictionary for arousal prediction.

Exponential averaging

The next step is to change the simple averaging to exponential averaging to introduce some continuity and smoothing of the emotion labels. The predicted label is calculated as the weighted average between the current average (over words found in a dictionary) and the predicted value for the previous second, with the exception of the first second where lyrics appear, where only the average is used.

Table 6.6: RMSE results for the exponential averaging technique showing various coefficients with both affective norms dictionaries, standard and short metric

	ANEW				Warriner			
	Arousal		Valence		Arousal		Valence	
Coefficient	RMS	s-RMS	RMS	s-RMS	RMS	s-RMS	RMS	s-RMS
0.9	0.352	0.315	0.318	0.292	0.438	0.400	0.311	0.291
0.7	0.350	0.313	0.308	0.283	0.428	0.389	0.294	0.274
0.5	0.350	0.312	0.301	0.275	0.418	0.378	0.281	0.260
0.3	0.350	0.311	**0.296**	**0.269**	0.408	0.367	0.275	0.251
0.2	0.350	0.311	**0.295**	**0.268**	0.404	0.362	**0.274**	**0.249**
0.1	0.351	0.311	**0.295**	**0.268**	**0.401**	**0.359**	0.278	0.252

The results achieved using different coefficients for the exponential averaging can be seen in Table 6.6, as RMSE is more informative at this point, only standard and short RMSE are noted in the table. The gap between the two dictionaries is starting to change: it is decreasing for the arousal axis and increasing for the valence axis,

especially with lower coefficient values. We can see that exponential averaging results in a substantial improvement in performance—there is little change in the results for the arousal axis using the ANEW dictionary, but the other three models are improved by close to or over 10%. The fact that the results tend to improve as the coefficient gets smaller show the need for continuity between subsequent labels and the importance of the temporal information. It is important to note, though, that the results seem to plateau at 0.1-0.3, with the optimal results (at least for the valence axis) achieved at 0.2.

Song and second averages

Now, since we know that adding some smoothing and temporal information to our labels helps, and given the lack of words in some samples (but not lack of emotion), the logical next step is to include the information of the overall emotion average. This also relates to the relative feature representation, which was useful for machine learning models using audio features. The song average is calculated as the average emotional value of all words that are found in the affective norms dictionary used. The predicted label is then calculated as the weighted average between the average (over words that are found) for a particular second and the overall average

$$\text{predicted\_label} = \alpha * \text{second\_average} + (1 - \alpha) * \text{song\_average}$$

Table 6.7: RMSE results for the weighted average between song and second averages using various coefficients with both affective norms dictionaries, standard and short metric

	ANEW				Warriner			
	Arousal		Valence		Arousal		Valence	
Coefficient	RMS	s-RMS	RMS	s-RMS	RMS	s-RMS	RMS	s-RMS
0.9	0.350	0.313	0.312	0.285	0.442	0.402	0.306	0.285
0.8	0.349	0.311	**0.307**	0.278	**0.440**	0.398	0.292	0.269
0.7	**0.348**	0.309	**0.307**	**0.273**	**0.440**	0.395	0.281	0.255
0.6	**0.348**	0.307	0.313	**0.274**	0.442	**0.393**	0.272	0.242
0.5	0.349	0.306	0.324	0.278	0.444	**0.392**	0.265	0.231
0.4	0.351	**0.305**	0.339	0.287	0.447	**0.393**	0.260	0.222
0.3	0.353	**0.305**	0.359	0.298	0.451	0.394	**0.258**	0.214
0.2	0.357	0.306	0.382	0.313	0.456	0.397	**0.259**	**0.209**
0.1	0.360	0.307	0.408	0.331	0.462	0.402	0.263	**0.208**
0.0	0.365	0.309	0.436	0.352	0.470	0.407	0.269	0.210

Results in Table 6.7 are shown for the full range of the coefficient values. We see a similar effect to that seen in the results for exponential averaging, except that now the dip (or the peak) happens for all models, though at completely different points. ANEW-based model results peak at 0.4-0.7 (depending on weather we look at the long or short RMSE, and depending on the axis), and Warriner based arousal models peak at 0.5-0.8. The effect of song and second averaging varies a lot depending on the dictionary and the axis—ANEW valence and Warriner arousal models perform worse than with exponential averaging, while the ANEW arousal

model is improved slightly and the Warriner valence model is improved by quite a lot. The last row (and the worsening of results as we are approaching it) shows once again that a simple song average does not suffice and second-by-second variation is important, but should not be used as is.

Song and exponential averages

The final “rule based" approach is now to combine the overall song average and the exponential averaging of each second. We now have two parameters whose values we need to pick, as in addition to the coefficient for the exponential averaging we have a coefficient for the weighted average between the overall emotion label and the exponential averaged emotion label for a particular second. For brevity, only the short RMSE results are reported—long RMSE results follow a similar trend, and short RMSE results should be considered more important as they reflect per-song performance of a model.

Table 6.8: Short RMSE results for the weighted average between song and exponential averages using various coefficients and Warriner affective norms dictionary

		Weighted average								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Arousal										
Exponential c.	0.9	0.407	0.401	0.397	0.393	0.391	0.390	0.390	0.391	0.393
	0.7	0.407	0.401	0.395	0.391	0.388	0.385	0.384	0.384	0.385
	0.5	0.407	0.400	0.394	0.389	0.384	0.381	0.378	0.377	0.376
	0.3	0.407	0.399	0.392	0.385	0.380	0.376	0.372	0.370	0.368
	0.2	0.407	0.399	0.391	0.384	0.378	0.373	0.369	0.366	0.364
	0.1	0.407	0.398	0.390	0.382	0.376	0.371	0.367	0.364	**0.361**
Valence										
Exponential c.	0.9	0.210	0.207	0.208	0.212	0.219	0.227	0.237	0.249	0.262
	0.7	0.210	0.207	0.206	0.209	0.214	0.220	0.228	0.238	0.249
	0.5	0.210	0.206	0.205	0.206	0.209	0.214	0.221	0.229	0.238
	0.3	0.210	0.206	0.204	0.204	0.206	0.210	0.216	0.222	0.231
	0.2	0.210	0.205	**0.203**	**0.203**	0.205	0.209	0.214	0.221	0.229
	0.1	0.210	0.205	**0.203**	**0.203**	0.205	0.209	0.215	0.222	0.231

Table 6.8 shows the results for a range of exponential coefficient values used—as in the case where only the exponential averaging is used, the results peak at lower exponential coefficient values (0.1-0.4) and then get worse as the coefficient is increased. The optimal averaging coefficient depends on the axis—the arousal model performs best with high averaging coefficient, i.e. when the song average is mostly ignored, while the valence model peaks with lower coefficient values (0.3-0.4). The combination of the two techniques achieves the best valence results (0.203) while arousal results are comparable, although slightly worse, than the best results achieved so far with simple exponential averaging (0.361 compared to 0.359).

Results for the ANEW dictionary show similar features to those of Warriner dictionary—they are extremely stable for the exponential coefficient values 0.1-0.4, and

the results get worse with higher coefficient values. The best results (0.250 for valence and 0.304 for arousal) are achieved with a lower coefficient (0.4-0.6) for arousal and higher value (0.9) for valence. Arousal values are comparable to those achieved with simple song and second averaging (0.304 compared to 0.305), while valence results are the best achieved with ANEW so far.

LDA

The first step when analysing LDA models is to look at the top words in its topic model, which gives an insight into both the training data used and the quality of a model that is achieved.

The AP-trained LDA gives a set of fairly defined topics that are generally quite political and geographical. E.g.:

- court, judge, case, charges, attorney, trial, prison, federal, state, drug
- space, souter, court, i, shuttle, telescope, mission, ms, two, nasa
- aids, health, drug, disease, researchers, virus, patients, research, blood, system
- soviet, east, union, german, west, gorbachev, germany, soviets, moscow, united

The MXM100 and MXM datasets result in much less well defined set of topics. While some of them still resemble a topic of some sort, a lot of the others seem to be a collection of meaningless words that tend to co-occur.

Examples from MXM100 model:

- are, get, die, you, dream, good, a, blood, sure, street
- you, me, a, and, to, do, my, not, is, now
- wait, real, done, door, fear, who, one, me, push, my
- word, time, me, wait, there, las, men, cuando, empti, keep

Examples from MXM model:

- that, to, you, my, we, and, a, thing, what, no
- wait, real, door, done, fear, who, one, me, town, my
- la, wonder, blood, sure, shame, sempr, taught, fi, santa, gold
- far, listen, ho, me, ago, polit, promis, and, victori, excus

The easiest way of using LDA models is to infer the topic distributions of a text or, in this case, the set of words occurring in each second, and use them as feature vectors for training a machine learning model on top of it. Table 6.9 shows the results of such an experiment using SVR with an RBF kernel. The first observation one can make is that the results achieved using this model are much better than any of the other lyrics-based models so far. With this basic use of LDA, there does not seem to be any real difference between the three datasets, and the results are nearly the same based on all four metrics.

Table 6.9: Results for the SVR model trained on topic distributions of the words occurring in each second, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
AP	0.330	0.286	0.002	-0.008	0.243	0.205	0.005	0.011
MXM100	0.328	0.287	0.026	-0.010	0.241	0.205	0.020	0.032
MXM	0.327	0.286	0.021	-0.058	0.242	0.206	0.020	0.001

When CCNF is used instead of SVR, we see a small expected improvement in the overall results (Table 6.10). What is more surprising is that CCNF reveals the difference between the three LDA models even with the basic feature vector. We can see a marked improvement of using LDA trained on MXM or MXM100 datasets when compared to one trained on the AP dataset. There does not seem to be much of a difference between MXM and MXM100 trained LDA models.

Table 6.10: Results for the CCNF model trained on topic distributions of the words occurring in each second, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
AP	0.332	0.286	0.002	-0.034	0.247	0.206	0.079	0.041
MXM100	**0.318**	**0.274**	**0.073**	-0.059	0.239	0.202	0.115	0.057
MXM	0.321	0.278	0.068	-0.117	**0.232**	**0.195**	**0.131**	0.062

As we have seen from the use of affective norms dictionaries, including a song average can improve the results, so in the next set of experiments, in addition to the topic distribution for the words occurring in the current sample, I also include the topic distribution for the words occurring in the whole extract. Table 6.11 shows the results achieved by the SVR model using such feature vectors. Although the improvement from the model described above is not big, it is definitely evident that including the topic distribution for the whole extract improves the results achieved by the models. We are also now starting to see the difference between the AP and the MXM datasets when using SVR—LDA models trained on both MXM datasets perform better than the ones trained on the AP dataset. It is especially notable with the correlation results. A small difference between the MXM100 and MXM datasets also starts becoming apparent—there is not much difference in the arousal results, but the results for valence axis show a small improvement when the whole dataset is used, and it is especially visible with short RMSE and long correlation.

Table 6.11: Results for the SVR model trained on topic distributions of the words occurring in each second and of the words occurring in the whole extract, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
AP	0.330	0.283	0.035	0.000	0.242	0.204	0.028	0.000
MXM100	**0.325**	**0.280**	**0.063**	-0.013	0.236	0.197	0.089	0.031
MXM	**0.324**	**0.279**	**0.066**	-0.054	**0.234**	**0.192**	**0.144**	0.026

When CCNF is used to train the model, we do not see a consistent improvement in the results (Table 6.12)—in fact, there is either no improvement at all, or the results are considerably worse than both the same feature set trained with SVR and the previous CCNF model. The ranking between the three LDA models is also less clear now—while the MXM-based model still outperforms the AP-based model with all metrics, the ranking between AP and MXM100 datasets is flipped when using long RMSE. The ranking between MXM and MXM100 datasets is also inconsistent, and it is flipped when using short RMSE. It would appear that with CCNF we are now starting to see the deteriorating effects of the increased vector size when using a small dataset to train on.

Table 6.12: Results for the CCNF model trained on topic distributions of the words occurring in each second and of the words occurring in the whole extract, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
AP	0.344	0.292	0.032	-0.045	0.253	0.211	0.015	0.064
MXM100	0.345	**0.283**	0.040	-0.023	0.260	0.211	0.068	**0.097**
MXM	**0.331**	0.288	**0.059**	0.000	**0.241**	**0.191**	**0.136**	0.047

Combined approach

The final model based on lyrics analysis only is a machine learning model trained on a feature vector that combines all of the approaches mentioned above. First of all, for both axes, it includes the simple affective norms averages for all three dimensions based on the Warriner affective norms dictionary. It also includes a song average for the three dimensions and the exponential averaging (using the best performing coefficient 0.2) for all dimensions using the same dictionary. I chose not to include the weighted average between the sample average and the song average, as I expected the machine learning model to learn that relationship. The feature vector also includes the topic distributions for both the set of words appearing in a particular sample and the set of songs appearing in the whole extract.

Table 6.13 shows the results achieved by this approach when trained using SVR model. We can see that there is a small, but consistent improvement over all metrics (with a potential exception of short correlation which does not appear to be reliable for these set of experiments) when compared to a model trained on LDA features only. Models for both axes are improved, and the arousal models based on all three datasets are now performing at a similar level, while there still is a large gap between AP and the two MXM models for the valence axis.

The effect of adding affective norms features on the models trained using CCNF (Table 6.14) is similar to that on the SVR trained models. There is a small improvement observable with all three models and with most evaluation metrics. The effect on the arousal axis is stronger when measured with short RMSE, where it is nearly 2%, while for the valence axis, the effect is similarly strong with both short and long RMSE.

Table 6.13: Results for the SVR model trained on combined analyses of lyrics, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
AP	0.323	0.275	0.057	-0.039	0.235	0.198	0.084	0.012
MXM100	0.322	0.274	**0.102**	0.001	**0.225**	**0.188**	0.152	0.014
MXM	**0.320**	0.273	0.076	-0.039	0.228	0.190	**0.163**	0.029

Table 6.14: Results for the CCNF model trained on combined analyses of lyrics, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
AP	0.342	0.289	0.048	-0.108	0.245	0.205	0.044	0.091
MXM100	0.345	**0.282**	0.029	0.000	0.253	0.206	0.078	0.080
MXM	**0.331**	**0.283**	**0.070**	-0.133	**0.237**	**0.189**	**0.140**	0.054

6.2.4. Conclusions

In this section I have described two types of analyses of lyrics that can be adapted to work for continuous dimensional emotion prediction in music. Lyrics-based analysis can be expected to improve valence predictions, and through my experiments I have shown that such features achieve lower RMSE values for valence models than for arousal models, while the opposite is true for acoustic features based models.

The first type of features—extracted from affective norms dictionaries—can be used in a rules-based system as they can produce an affective prediction directly. In addition to the simple label averaging I suggested exponential averaging and averaging between second label and the song average as two techniques that can provide more continuity and smoothing of labels between different samples. Both types of techniques improved the results when compared to the basic approach: the combination of the two was the best approach for the valence model and it reduced long RMSE by up to 22.8% and short RMSE by up to 32.6%. For ANEW dictionary-based analysis of the arousal model, exponential averaging combined with song averaging was also the best approach, and it reduced long RMSE by 1.4% and short RMSE by 3.8%, while for the Warriner dictionary based arousal model exponential averaging was the most successful approach, reducing long RMSE by 9.7% and short RMSE by 11.8%.

The second type of analysis is done using Latent Dirichlet Allocation, which is an algorithm from a family of techniques used to reduce the number of dimensions used to represent a document and which achieve a semantic representation, or latent topic representation of text. In order to use it for continuous musical emotion prediction, I trained three different LDA models (one based on a general text corpus and two on corpora consisting of lyrics only). I extracted topic distributions from words occurring in a second, and also from all the words occurring in the whole extract and used them both to train two machine learning

models (SVR and CCNF). LDA-based machine learning models performed better than all the affective norms dictionary based techniques: the CCNF model using topic distributions of words occurring in a single second reduced long RMSE for arousal by 20.7% and valence by 6.1% and short RMSE for arousal by 23.7% and valence by 3.9%. Short RMSE for valence is further improved by including topic distribution of the whole extract into the feature vector, and the total reduction is 5.9%. The best model for valence is achieved by combining all the different lyrics features and using SVR for training: long RMSE is 0.225 (6.6% improvement over the simplest SVR model and 29.7% over simplest affective norms dictionary based model) and short RMSE is 0.188 (8.3% reduction from simplest SVR model and 37.5% over simplest affective norms dictionary based model). I have also shown that LDA trained on a corpus of lyrics will most of the time outperform a general LDA model, but even a general LDA model can achieve good results.

6.3. Fully multi-modal system

Having looked at all the individual parts that constitute a song, it is time to put them all together into one comprehensive system. The intuition is clear—there is no doubt that when listening to music we identify all three modalities (instruments, vocals and lyrics), and that we extract and expect different features from them. A system that attempts to do the same, one would hope, should achieve better results than one which ignores one or more of the modalities.

The system for multi-modal continuous dimensional emotion recognition in music that I am describing in this section relies on the work described in both the previous chapters as well as previous two sections. The best feature-vector-building techniques are combined with the best voice-extraction technique and with the most promising lyrics analysis techniques.

6.3.1. Methodology

As none of the voice and music separation techniques clearly outperformed any of the others, here I am using two that were often the best in a set—REPET-SIM and VUIMM. The audio files were analysed using OpenSMILE, and the same features were extracted as those used elsewhere in this work.

For the lyrics analysis, both types of features are used: affective norms dictionary-based features and LDA features. For the affective norms dictionary-based features, I am using the Warriner affective norms dictionary and simple averaging of all dimensions both for each second and over the whole song, as well as exponential averaging using the coefficient of 0.2. The LDA-based features include topic distributions of both the set of words appearing in a second and the set of words appearing in the whole song. All three LDA training tests are used here, and LDA underwent the same training process as described in Section 6.2. All the lyrics-based features used in this section are described in Section 6.2.

The feature vector is constructed using simple feature fusion, by concatenating all the different features into one feature vector. Two feature representation techniques are used in these experiments: basic and relative. The relative feature rep-

resentation (Section 4.7) used here is slightly modified from the original—acoustic features are represented as before in the relative feature representation, and lyrics features are represented using song averages calculated as part of lyrics analysis, rather than simple feature averages as in the original.

The dataset used in this section is once again the MoodSwings dataset. While the analysis of lyrics could be done on the full dataset, the actual audio files are necessary for the voice extraction, meaning that a smaller subset of MoodSwings (that includes only the songs whose audio files I have and those containing lyrics) is used here, as described in Section 6.2.2.

The same experimental design as elsewhere in the dissertation is used in the experiments described in this section. 5-fold cross-validation is used in all of the experiments with the same distribution of songs between the 5 folds used in all of the experiments—the distribution is also the same as the one used in Section 6.2, as these are the only two sections using the exact same dataset. 2-fold cross-validation is used within each training set to pick the best hyper-parameters, which are then used to train a model on the whole training set. This way the training samples are always completely separate from the testing samples and the risk of over-fitting is minimised. All of the results are calculated per fold and the average results over all folds are reported.

6.3.2. Results

For the purpose of having a proper baseline, I also retrained both SVR and CCNF with a feature vector composed of only the acoustic features extracted from the original audio signal. While it would be possible to simply compare the results achieved here with those described in Chapters 4 and 5, neither the dataset, nor the song distribution would be exactly the same, and so the results would be somewhat less comparable.

As elsewhere, four metrics are used here: short and long RMSE, long correlation and short non-squared correlation. Lower RMSE results are considered better, and the opposite is true about correlation results.

Full multi-modal system

The starting point of my experiments was to use the complete set of acoustic and lyrics-based features.

The fully multi-modal model trained on SVR (Table 6.15) seems to be performing better than the same model using only acoustic analysis of the original songs. For the arousal axis, REPET-SIM with MXM100-trained LDA features seems to achieve the best results, but all models using REPET-SIM perform better than the simple original audio-based arousal model. For the valence axis, REPET-SIM also seems to be the more suitable separation technique with all three models generally improving the results when compared to the acoustic model. VUIMM based models are performing worse overall, but both MXM models are still an improvement over the acoustic model with MXM100 producing the best results.

A quick look at Table 6.3 suggests that the addition of lyrics-based features also improves the results when compared with the bi-modal feature analysis. While the results of the arousal model are comparable and the analysis of lyrics do not seem to make much of a difference, the results of the valence models are definitely improved.

Table 6.15: Results for REPET-SIM and VUIMM music-voice separation techniques combined with lyrics analysis using AP, MXM100 and MXM datasets, compared with original acoustic analysis and the same separation techniques without the analysis of lyrics, SVR with RBF kernel, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
Acoustic	0.202	0.185	0.625	0.011	0.213	0.186	0.227	0.071
REPET-SIM								
AP	**0.184**	**0.163**	**0.688**	-0.006	0.212	0.181	0.243	0.048
MXM100	0.199	0.176	0.637	0.012	**0.206**	**0.173**	**0.292**	0.057
MXM	0.195	0.172	0.660	0.006	0.217	0.183	0.256	0.065
VUIMM								
AP	0.198	0.180	0.640	0.029	0.218	0.186	0.196	0.043
MXM100	0.204	0.184	0.623	0.044	0.206	0.176	0.283	0.052
MXM	0.202	0.182	0.625	0.047	0.214	0.182	0.245	0.045

When the same feature vectors are used to train CCNF models, we see a completely different image. Several signs indicate that CCNF is struggling to find meaningful patterns in the expanded feature vectors. First of all, we see large differences between the three models of LDA—much larger than it would be expected. Moreover, none of the multi modal models manage to outperform the simple acoustic model. Finally, the variation of results between different folds is increased. All of these suggest that CCNF is unable to cope with the increased feature vector size—if the addition of the lyrics features provided no useful information we would see no change in results (when compared to model trained on two modalities only). The only interesting point about the results achieved by CCNF using the full multi-modal model is the increased short correlation of the valence models. All 6 models show fairly high short correlation, which is between 31% and 71% bigger than that for the simple acoustic model, and which is also not present in the SVR-trained models.

Reduced multi-modal system

As CCNF was clearly struggling with the expanded feature vector, I chose to reduce the set of features somewhat by not including the LDA-produced topic distributions for the whole extract. In the LDA-only experiments, the addition of this set of features was not convincingly beneficial to the model, and I suspected that removing 50 features from the feature vector might be more beneficial than including these features, especially for the CCNF model. In these experiments I only used REPET-SIM separation, as this technique generally produced better results than VUIMM in the fully multi-modal experiments.

Table 6.16: Results for REPET-SIM and VUIMM music-voice separation techniques combined with lyrics analysis using AP, MXM100 and MXM datasets, CCNF, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
Acoustic	**0.173**	**0.150**	**0.731**	0.025	**0.200**	**0.165**	**0.335**	0.073
REPET-SIM								
AP	0.175	0.150	0.719	0.038	0.231	0.186	0.186	**0.125**
MXM100	0.232	0.172	0.527	0.000	0.233	0.184	0.203	0.098
MXM	0.184	0.153	0.688	0.000	0.222	0.178	0.290	0.101
VUIMM								
AP	0.187	0.159	0.679	0.059	0.226	0.182	0.200	0.107
MXM100	0.234	0.174	0.519	0.000	0.223	0.179	0.216	0.103
MXM	0.191	0.160	0.670	0.000	0.216	0.173	0.311	0.096

SVR trained on the reduced feature vector (Table 6.17) performs similarly well or improves the results of the best performing multi-modal models. Results of lower-performing LDA models from the full multi modal system (Table 6.15) are improved and the performance of all three LDA models is now near identical. All three models now perform substantially better than a simple acoustic model for both axes.

Table 6.17: Results for REPET-SIM music-voice separation techniques combined with reduced lyrics analysis using AP, MXM100 and MXM datasets, SVR with RBF kernel, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
AP	0.183	0.165	0.694	-0.018	0.205	0.178	0.292	0.059
MXM100	0.181	0.163	0.700	0.004	0.206	0.178	0.292	0.058
MXM	0.182	0.165	0.696	-0.011	0.205	0.177	0.294	0.055

With CCNF, we see a marked improvement when the song topic distribution is removed from the feature vector and only the second topic distribution and the affective norms dictionary features remain. Results of the three LDA modes (Table 6.18) are similar (especially with the arousal models, where the results are nearly identical), just like with the SVR models. We now see an improvement over a simple acoustic model trained with CCNF for arousal, and valence results of the two methods are now comparable. An interesting feature of this model is the much higher short correlation achieved by all the valence models—there is a substantial improvement over all the previous methods (apart from the moving relative and delay window feature representations).

Relative feature representation

It was also interesting to see if using relative feature representation could improve the results of a fully multi modal system as much as it improves the results of a simpler system relying on only acoustic features. For these models I modified

Table 6.18: Results for REPET-SIM music-voice separation techniques combined with reduced lyrics analysis using AP, MXM100 and MXM datasets, CCNF, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
AP	0.164	0.141	0.745	0.014	0.204	0.168	0.303	0.138
MXM100	0.166	0.141	0.745	0.031	0.200	0.164	0.328	0.145
MXM	0.164	0.141	0.754	0.043	0.200	0.166	0.330	0.129

the relative feature representation slightly: for LDA features, I used song topic distribution as the song average included in every feature vector, and I used the difference between that and the topic distribution for a particular sample as the relative feature; for affective norms based features, I used song average instead of average feature that is included in every feature vector and only the simple sample average and the song average difference as the relative feature; I omitted the exponential average features completely, as they did not seem to fit with the rest of the scheme.

As CCNF was already struggling with increased feature vector size and generally does not benefit from relative feature representation, I only used SVR for these experiments. The addition of lyrics features in the relative representation seems to improve a model based on only on REPET-SIM features using relative feature representation. For arousal, AP-trained features were the most beneficial reducing RMSE and increasing correlation, while valence benefited the most from MXM100 set of features, achieving the best SVR-trained valence model results reported in this dissertation (except for short correlation results).

Table 6.19: Results for REPET-SIM music-voice separation techniques combined with lyrics analysis using AP, MXM100 and MXM datasets and relative feature vector representation, SVR with RBF kernel, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
REPET-SIM	0.182	0.152	0.696	0.017	0.212	0.175	0.273	0.044
AP	**0.167**	**0.142**	**0.746**	0.024	0.208	0.171	0.273	0.013
MXM100	0.179	0.153	0.712	0.026	**0.203**	**0.167**	**0.311**	0.029
MXM	0.174	0.148	0.727	0.024	0.207	0.171	0.290	0.033

Multi-modal model with song-only lyrics features

The final set of experiments is my attempt to show that the analysis of lyrics could be beneficial to continuous dimensional emotion tracking even when automatic singing voice transcription and the exact timing of words is not available. Automatically acquiring the lyrics for a whole song is reasonably easy and there are systems which provide such a service, which means that the features which are extracted from lyrics for the whole song rather than for each second could still be used. The feature vector used in the experiments described in this section contains all the acoustic features extracted from both modalities, but only the song averages

extracted from the affective norms dictionaries and the LDA topic distributions for the whole extract.

While the performance of the best SVR model trained on such a vector (AP for arousal and MXM100 for valence) is as good as (or slightly better for arousal than) the best models with a reduced multi modal vector and the full multi modal system, the overall results are slightly worse than with the reduced feature vector. Despite that, all the models still outperform the basic model with simple acoustic analysis.

Table 6.20: Results for REPET-SIM music-voice separation techniques combined with song-only lyrics analysis using AP, MXM100 and MXM datasets, SVR with RBF kernel, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
AP	**0.181**	**0.161**	**0.700**	-0.004	0.211	0.179	0.244	0.069
MXM100	0.192	0.171	0.666	0.016	**0.206**	**0.173**	**0.291**	0.054
MXM	0.191	0.169	0.668	0.017	0.213	0.180	0.269	0.066

Unlike with the SVR models, CCNF models does not seem to benefit at all from the addition of song-average lyrics features (Table 6.21). All of the results achieved with these models closely resemble results achieved with REPET-SIM and full lyrics features (Table 6.16), while being slightly worse than the original. Since the reduced lyrics feature vector improves CCNF results, it would seem that CCNF struggles to find meaningful patterns in song topic distributions, while SVR manages to improve the model.

Table 6.21: Results for REPET-SIM music-voice separation techniques combined with song-only lyrics analysis using AP, MXM100 and MXM datasets, CCNF, standard and short metrics

	Arousal				Valence			
	RMS	s-RMS	Corr	s-Corr	RMS	s-RMS	Corr	s-Corr
AP	0.177	0.150	0.714	0.060	0.234	0.189	0.176	0.122
MXM100	0.234	0.175	0.517	0.000	0.234	0.185	0.194	0.089
MXM	0.186	0.154	0.685	0.000	0.219	0.176	0.298	0.066

6.4. Conclusions

In this chapter I have shown three main things: that separating vocals and background music and analysing them separately can improve arousal results; that sentiment analysis in text can be adapted to work with continuous dimensional emotion prediction in music and that it achieves better valence results than arousal; and, finally, that a fully multi-modal system can improve the performance of a continuous dimensional music emotion prediction system.

Firstly, I have tried four different music and vocals separation techniques, and compared the suitability of such audio for emotion prediction in music. When only the music part of the original audio is used, it can improve the performance of the arousal models (reducing long RMSE by up to 8.7%, short RMSE by

11.7%, and increasing long correlation by up to 10%, with stronger improvement for standard feature representation, and more for SVR rather than CCNF models). The performance of valence models seems to suffer a bit from the lack of vocals and both models seem to suffer a lot by the lack of background music (when only the vocals are used). When the features from the two modalities are combined into a single feature vector, it brings the performance of valence models back to its original level, and continues to improve the performance of arousal models. This dual analysis seems to benefit CCNF trained models more than the SVR trained models and we now start to see the difference between the two when the relative feature representation is used. The long RMSE is decreased by up to 8.2%, short RMSE by up to 13.2%, and long correlation is increased by up to 10.8% for arousal.

Secondly, I have adapted two families of sentiment analysis in text techniques (based on affective norms dictionaries and on semantic representation of terms) to work for continuous dimensional emotion recognition, also explaining why some of the other common IR techniques might not be suitable for such a task. Affective norms dictionaries can be used to produce a rule-based emotion recognition system—I suggested the use of simple averaging of the labels for words appearing in a sample, an exponential averaging, and a weighted averaging between the those and a song average. Using two different affective norms dictionaries (Warriner and ANEW) I showed that exponential averaging or weighted averaging generally improves the results (showing the need for continuity), and that the Warriner dictionary (which is over 10 times bigger than ANEW) produced better valence results. The second family of features are derived from a semantic representation of terms occurring in a body of text—in this thesis I used LDA and trained it on three different datasets (one general corpus, one containing a large set of lyrics and another containing a subset of those). The topic distributions of words can then be used as a feature vector for a machine learning model. In my experiments I tried three different models: one containing only the topic distributions of words occurring in one second, a combination of words occurring in a second and in the whole extract and one containing the two sets of topic distributions and the affective norms dictionary based features. The last model trained with SVR was shown to achieve the best valence results: long RMSE is 0.225 (6.6% improvement over the simplest LDA based SVR model and 29.7% over simplest affective norms dictionary based model) and short RMSE is 0.188 (8.3% reduction from simplest SVR model and 37.5% over simplest affective norms dictionary based model).

Finally, I have shown that the three modalities (music, vocals and lyrics) can be combined to produce a model performing better than the one based on acoustic analysis only. CCNF was generally struggling with the increased feature vector size, and CCNF-based models achieved only mediocre results, while SVR trained models benefited a lot from the combination of the three modalities. When compared with an acoustic model, a fully multi-modal model was able to reduce arousal long RMSE by up to 8.9%, short RMSE by up to 11.9%, and increase long correlation by up to 10.1%; for valence the long RMSE was reduced by up to 3.3%, short RMSE by 7.0%, and long correlation increased by up to 28.6%. The reduced feature vector (lacking song topic distributions) managed to improve the results a bit further, but the best performing model was the one using relative feature rep-

resentation: the total improvement over an simple acoustic model for arousal was a 17.2% reduction in long RMSE, 23.2% reduction in short RMSE, and 19.4% increase in long correlation; for valence it was a 4.7% reduction in long RMSE, 10.2% reduction in short RMSE and a 37.0% increase in long correlation. While these results were achieved with second-specific lyrics features, which require manual tagging, a model using only song-specific lyrics features, which could be extracted automatically, was shown to also bring an improvement over an acoustic model. The models containing features extracted from LDA trained on general corpus were often the best performing models for the arousal axis, while the lyrics-based corpora were more suitable for valence models.

Conclusion

7

7.1. Contributions

The contributions of this dissertation can be grouped into three groups, each describing one aspect of a machine learning solution to the problem of automatic continuous dimensional emotion prediction in music: evaluation (and the evaluation metrics), the machine learning model and the feature vector used, and multi-modal data analysis.

7.1.1. Evaluation metrics

As continuous dimensional emotion recognition in music is a new field, there are no agreed-upon metrics used to evaluate the different approaches, and various techniques are used to report the results achieved by different researchers. During my work I designed and executed a novel study which attempted to determine people's intuitively preferred evaluation metric. The results of the study provide evidence that distance-based metrics might be more preferable to correlation in one-dimensional cases, but no single metric can capture all the different aspects of a problem, and, ideally, several metrics should be reported (Chapter 3). In addition to that, throughout this dissertation I repeatedly showed how big an effect the choice between even the most popular evaluation metrics has on the reported performance or ranking of various approaches.

7.1.2. Balance between ML and feature vector engineering

Another important contribution of my work is the introduction of two new machine learning models that have never before been used for a continuous emotion prediction in music (Chapter 5). Both models encode some of the temporal information that is otherwise lost in general bag-of-frames approaches to the problem. In addition to that, I have proposed several novel feature representation techniques that are based on various findings in the general field of Emotion in Music, which bring various improvements to the results as measured by different evaluation metrics (Chapter 4). An important conclusion of this work is that the same level of improvement can be brought by either a more complex feature representation

technique or a more complex machine learning model, and that the combination of the two does not necessarily improve the results.

7.1.3. Multi-modality

Finally, I have enriched my solution to continuous emotion prediction in music by using multi-modal analysis of songs. While there exist some approaches using multi-modal data to predict static emotion in music, there are not many, if any, solutions to the continuous version of the problem. In my work I have used separation of the vocals and music, and analyzed them separately, as well as modified some techniques of sentiment analysis in text to make them work in the continuous case. As both types of analysis depend on techniques for problems that are not solved yet (separation of vocals and music, and automatic singing voice transcription), the final system is a proof-of-concept showing the potential benefit, rather than a fully functioning system that utilises it (Chapter 6).

7.2. Limitations and future work

While I have made some contributions to the field of continuous dimensional emotion recognition in music, the problem is by no means solved. The following sections identify the major weaknesses of my approach and suggest some future work directions.

7.2.1. Online data

One of the most important parts of a machine learning solution is its training data. It has to be representative, clean and reliable, as well as reasonably big—violation of any one of these requirements would greatly disadvantage the system and lead to inferior results. Unfortunately, there is not a big choice of publicly available datasets for continuous dimensional emotion recognition yet, nor is there a standard dataset that everyone uses. While the datasets available are getting increasingly large, and more varied, they are mostly collected using an online tool (usually Amazon Mechanical Turk), without any or much control over what participants participate, or what the conditions when they take the study are. This inevitably leads to introduction of a lot of noise in the labels—when over 32% of all labelers can have all of their responses automatically rejected because they are obviously fake [Speck et al., 2011], one can only wonder what the percentage of unreliable labels remain in the datasets we use.

In the ideal scenario we would have a large dataset of music (including the recordings) with longer extracts that was annotated in a laboratory with carefully controlled experimental conditions and a good experimental design. For as long as that is not the case we can only hope that a large number of annotations that can be quickly collected online results in a reliable and representative average.

7.2.2. Popular music

The reliance on popular music datasets is one of the advantages as well as drawbacks of this work. While I made a conscious choice to use popular music for my

experiments, therefore making the system more applicable to the everyday needs of people, it might limit the system's capabilities. It remains to be seen if a system trained on popular music can successfully be used to classify classical music—the underlying musical features are the same, or similar, but their range is probably a lot more limited in popular music. It might also be the case that a system trained on classical music, and therefore exposed to a wider range of patterns would be superior to one trained on popular music only. But as it is often the case with machine learning solutions, they perform better when they are more specialised, so I suspect that this system should only be used for Western popular music, and extra considerations should be taken for other types of datasets.

7.2.3. Current dataset

Throughout this dissertation in each chapter we have seen the results plateauing, and each additional technique achieving a smaller improvement. While it is entirely likely that I have exhausted the benefits that could be gained with those particular approaches, another explanation is that this is as much as can be achieved with the dataset I was using. As the corpus is quite small and the extracts are quite short, it is possible that I have reached the best results that can be achieved using that dataset.

7.2.4. Imperfection of analysis

A lot of the work described in this dissertation falls under the heading "proof of concept" rather than a working solution. While I believe that the techniques I have described are sound and could produce good results, without the improvement in the following areas, a "perfect" system for automatic emotion recognition in music cannot be built. On a more positive note, I would expect a "free" improvement in the results simply by replacing the underlying methods that my system depends on.

Manual tagging of lyrics

The need to manually tag the lyrics is obviously the main hurdle against using sentiment analysis techniques for continuous emotion prediction in music. Before this problem gets solved, it is going to be impossible to incorporate most of these techniques into fully automatic systems. Several solutions or shortcuts exist. A large set of songs already have their lyrics transcribed and tagged with time information–karaoke versions of lyrics. While such songs still require manual work, they vastly increase the number of songs whose lyrics can be analysed on a second by second basis. Another shortcut is to include the whole-song information extracted from lyrics—while again we require someone to transcribe the lyrics of a song, we now have access to a lot more data and we can still extract features representing the whole mood of a song to improve continuous emotion detection in a song.

Music and voice separation

We run into a similar problem when thinking about the vocals-music separation. Unlike with the lyrics, this can be done automatically, but the quality of the separated audio is not outstanding. The noise present in the signal must affect the features extracted from the songs and therefore affect the results. There is a simple solution that does not involve waiting for the state-of-the-art algorithms to improve—using audio where the vocals and the background are separated into different channels. While it's unlikely that the general public will get access to songs distributed in such a way, if the benefit from having these modalities separate becomes established, the recording companies might give access to these to companies providing music to people.

Low-level features

Once again, the algorithms we are relying on for feature extraction, or lack of them, prevents us from using higher-level features for emotion prediction in music. There is a lot of well established research linking various musical features with various emotions, and one would expect to get good results with a system that relies on such findings. Unfortunately, the algorithms for extracting these are not established yet, so we have to resort to lower-level features that are easier to define and extract. Similarly to how karaoke versions of lyrics can be used to shortcut the problem of automatic transcription of singing voice, music sheet analysis could be used for the extraction of high-level musical features.

7.2.5. Future work

One of the most important steps towards a better emotion recognition system would be a better dataset that has been annotated in laboratory conditions rather than online. While I have already talked about the advantages of such a dataset, there is an addition to the set of songs that I think could lead to interesting research—songs where the mood of the lyrics disagrees with the mood expressed by the music. Besson et al. [1998]; Thompson and Russo [2004]; Ali and Peynircioglu [2006] have shown the effect that the disagreement between music and lyrics can have—it would be interesting to reproduce the effect with a larger study and it would also be a perfect opportunity to showcase the advantages of a multi-modal system, which takes into account both the music and the lyrics. It would also be interesting to see if there are any differences between native speakers and fluent speakers of a particular language that a song is sung in—this might have implications for the systems used in real life.

Throughout the various approaches described in this dissertation we have seen a marked difference between the effects on and of the two axes. While some of the approaches greatly improve the results for arousal models, they might have little impact for the valence models, and the opposite is true as well. These findings suggest that the two axes possibly require two different approaches, potentially using different machine learning models, using different modalities or feature representation techniques. It would be interesting to see if a better model could be built by

either treating these two axes completely separately, or, on the other hand, using machine learning models that can model both axes at the same time, therefore exploiting the interaction between them.

7.3. Guidelines for future researchers

At my PhD oral examination, my examiners asked me to consider adding to the dissertation some advice for future researchers in the field. With their post-exam approval, the following brief points summarise the advice I would give based on lessons that I have learned throughout my PhD. I trust that these are useful to anyone starting or considering research in this field.

- Start by reading a book or a collection of articles that describe the problem from a multi-disciplinary point of view. It is a multi-disciplinary problem, and it would be a shame to treat it as anything else. It will give you a better perspective and offer better solutions.
- On a similar note, look at the solutions to similar problems. If a technique has been successfully applied to emotion recognition from voice or facial expressions, it might be just as successful for emotion recognition in music.
- A lot of the lowest-hanging fruits have been picked already, so think about more integrated approaches. A lot of work has been done on applying general-purpose techniques (machine learning approaches) to this problem, but better results might be achieved by looking at problem-specific techniques, and features in particular.
- Do not be tempted to report only one metric! Even, or especially, if you can show improvement with one of them, but not with all of them. Reporting multiple metrics will lead to a better understanding of the problem and of the solution and will result in better approaches and results in the future.
- Train the baseline (and your own) system the best way you can. Do not just pick the default settings, or have a good reason why you are doing it. Settings and training parameters matter. If I do not think you trained your baseline method well, I will not trust the improvement you showed.
- Also, similarly to that, be careful to not overfit your system—there are a few things we have learned about training music information retrieval systems, and if you are not following the same procedure, have a good reason.
- On the other hand, do not do something because everyone else is doing it. It is a new field and we are just figuring things out. We have fallen into poor habits because it was easier at a time, and we need someone to point them out, so we could get better. If you stick with it, others will follow eventually.
- Find a good dataset or, if you have time, collect your own. This will be one of the main limitations of your work—if your dataset captures only very simplistic variations of emotion, you will quickly reach the ceiling imposed by your dataset.

– Do not let the tools-do-not-yet-exist problem stop you. We are not building a standalone, fully functioning system, which means that you can still base your analysis on lyrics or high level musical features, for example, even if if you cannot extract them automatically from music.

Finally, be prepared to listen to everyone's (including your neighbour and your grandmother) expert opinion on how such a problem should be tackled and features you should look into. And most importantly, good luck and have fun!

Bibliography

S Omar Ali and Zehra F Peynircioglu. Songs and emotions: are lyrics and melodies equal partners? *Psychology of Music*, 34(4):511–534, 2006.

Anna Aljanaki, Yi-Hsuan Yang, and Mohammad Soleymani. Emotion in music task at mediaeval 2014. In *MediaEval Workshop*, 2014.

D Bainbridge, S J Cunningham, and J S Downie. How People Describe Their Music Information Needs : A Grounded Theory Analysis Of Music Queries. In *Proc. of ISMIR*, pages 221–222, 2003.

T Baltrušaitis, N Banda, and P Robinson. Dimensional Affect Recognition using Continuous Conditional Random Fields. In *IEEE FG*, 2013.

Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. 3D Constrained Local Model for Rigid and Non-Rigid Facial Tracking. In *CVPR*, 2012.

Tadas Baltrušaitis, Louis-Philippe Morency, and Peter Robinson. Continuous conditional neural fields for structured regression. In *European Conference on Computer Vision*, 2014.

S. Baron-Cohen, O. Golan, S. Wheelwright, and J. J. Hill. Mindreading: The interactive guide to emotions, 2004.

Mathieu Barthet and George Fazekas. Multidisciplinary Perspectives on Music Emotion Recognition: Implications for Content and Context-Based Models. In *International Symposium on Computer Music Modeling and Retrieval CMMR*, pages 19–22, 2012.

M Besson, F Faita, I Peretz, A M Bonnel, and J Requin. Singing in the brain: Independence of lyrics and tunes. *Psychological Science*, 9(6):494–498, 1998. ISSN 09567976.

Yves Bestgen. Can emotional valence in stories be determined from words? *Cognition & Emotion*, 8(1):21–36, 1994. ISSN 02699931.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022, May 2003. ISSN 1532-4435.

Jerry Boucher and Charles E. Osgood. The pollyanna hypothesis, 1969. ISSN 00225371.

BPI. Digital Music Nation. Technical report, 2013.

Margaret MM Bradley and PJ Peter J Lang. Affective Norms for English Words (ANEW): Instruction Manual and Affective Ratings. *Psychology*, Technical(C-1): 0, 1999. ISSN 10897801.

R A Calvo and S D'Mello. Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications, 2010. ISSN 19493045.

V R Carvalho and Chih-yu Chao. Sentiment Retrieval in Popular Music Based on Sequential Learning. *Proc. ACM SIGIR*, 2005.

Wei Chai and Barry Vercoe. Using User Models in Music Information Retrieval Systems. *Information Retrieval*, page 2p., 2000.

Chih-chung Chang and Chih-jen Lin. LIBSVM: a library for support vector machines. *Computer*, 2(3):1–39, 2001. ISSN 21576904.

Ze-Jing Chuang and Chung-Hsien Wu. Emotion recognition using acoustic features and textual content. *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, 1, 2004.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

Eduardo Coutinho, Felix Weninger, Björn Schuller, and Klaus R Scherer. The Munich LSTM-RNN Approach to the MediaEval 2014 " Emotion in Music " Task. In *MediaEval Workshop*, pages 5–6, Barcelona, Spain, 2014.

R Cowie, C Cox, JC Martin, A Batliner, D Heylen, and K Karpouzis. Issues in data labelling. 2011.

Roddy Cowie, Cian Doherty, and Edelle McMahon. Using dimensional descriptions to express the emotional content of music. In *ACII*, pages 1–6, 2009.

Frank M Diaz and Jason M Silveira. Music and affective phenomena a 20-year content and bibliometric analysis of research in three eminent journals. *Journal of Research in Music Education*, 62(1):66–77, 2014.

Harris Drucker, Chris J.C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. *Advances in Neural Information Processing Systems*, (x):155–161, 1996.

J Durrieu, Bertrand David, and Gaël Richard. A musically motivated mid-level representation for pitch estimation and musical audio source separation. *Selected Topics in Signal Processing, IEEE Journal of*, 5(6):1180–1191, 2011.

Fabon Dzogang, Marie-jeanne Lesot, Maria Rifqi, and Bernadette Bouchon-meunier. Analysis of texts' emotional content in a multidimensional space. *International Conference on Kansei Engineering and Emotion Research*, 2010.

T Eerola and J K Vuoskoski. A comparison of the discrete and dimensional models of emotion in music. *Psychology of Music*, 39(1):18–49, 2010. ISSN 03057356.

Tuomas Eerola, Olivier Lartillot, and Petri Toiviainen. Prediction of multidimensional emotional ratings in music from audio using multivariate regression models. In Hirata K and G Tzanetakis, editors, *Information Retrieval*, pages 621–626. ISMIR, 2009.

Paul Ekman and Wallace V Friesen. *The facial action coding system*. Consulting Psychologists Press, 1978.

Moataz El Ayadi, Mohamed S. Kamel, and Fakhri Karray. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587, 2011.

P Evans and E Schubert. Relationships between expressed and felt emotions in music. *Musicae Scientiae*, 12(1):75–99, 2008. ISSN 10298649.

Florian Eyben, Anton Batliner, Björn Schuller, Dino Seppi, and Stefan Steidl. Cross-corpus classification of realistic emotions–some pilot experiments. In *Proc. 3rd International Workshop on EMOTION (satellite of LREC): Corpora for Research on Emotion and Affect, Valetta*, pages 77–82, 2010.

Yuchao Fan and Mingxing Xu. MediaEval 2014 : THU-HCSIL Approach to Emotion in Music Task using Multi-level Regression. In *MediaEval Workshop*, pages 6–7, Barcelona, Spain, 2014.

Paul Randolph Farnsworth. The Social Psychology of Music. *Iowa State University Press*, 45(2):304, 1958. ISSN 00274321.

Yazhong Feng, Yueting Zhuang, and Yunhe Pan. Popular music retrieval by detecting mood. *Proceedings of the 26th ACM SIGIR*, 2(2):375–376, 2003a.

Yazhong Feng, Yueting Zhuang, and Yunhe Pan. Music Information Retrieval by detecting mood via computational media aesthetics. In *Proc of the IEEEWIC International Coference on Web Intelligence*, volume telligence, pages 1–7. IEEE, 2003b. ISBN 0769519326.

A Gabrielsson. Emotion perceived and emotion felt: Same or different? *Musicae Scientiae*, 2001-2(Special Issue):123–147, 2002.

Alf Gabrielsson and E Lindström. The role of structure in the musical expression of emotions. In Patrik N Juslin and John Sloboda, editors, *Handbook of music and emotion Theory research applications*, pages 367–400. Oxford University Press, 2010.

Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101 Suppl:5228–5235, 2004.

Michael Grimm and Kristian Kroschel. *Emotion estimation in speech using a 3D emotion space concept*. I-Tech, 2007.

Di Guan, Xiaoou Chen, and Deshun Yang. Music emotion regression based on multi-modal features. In *9th International Symposium on Computer Music Modeling and Recognition*, 2012.

Hatice Gunes, Mihalis A Nicolaou, and Maja Pantic. *Continuous Analysis of Affect from Voice and Face*, pages 255–291. Springer London, 2011. ISBN 978-0-85729-993-2.

Byeong-jun Han, Seungmin Rho, Roger B Dannenberg, and Eenjun Hwang. SMERS: Music emotion recognition using support vector regression. *Information Retrieval*, (Ismir):651–656, 2009.

D J Hargreaves and A C North. Experimental aesthetics and liking for music. In P N Juslin and J A Sloboda, editors, *Handbook of music and emotions theory research applications*, chapter 19, pages 515–547. OUP, 2010.

Christopher Harte, Mark Sandler, and Martin Gasser. Detecting harmonic change in musical audio. *Proceedings of the 1st ACM workshop on Audio and music computing multimedia AMCMM 06*, C(06):21, 2006.

Hui He, Jianming Jin, Yuhong Xiong, Bo Chen, and Wu Sun. Language feature mining for music emotion classification via supervised learning from lyrics. *Advances in Computation*, 5370:426–435, 2008. ISSN 03029743.

Xiao Hu and J Stephen Downie. Exploring mood metadata: Relationships with genre, artist and usage metadata. *Information Retrieval*, pages 67–72, 2007.

Xiao Hu and J Stephen Downie. Improving mood classification in music digital libraries by combining lyrics and audio. *Proceedings of the 10th annual joint conference on Digital libraries JCDL 10*, pages 159–168, 2010a.

Xiao Hu and J Stephen Downie. When lyrics outperform audio for music mood classification: a feature analysis. In J Stephen Downie and Remco C Veltkamp, editors, *Proceedings of ISMIR*, number Ismir, pages 619–624, 2010b.

Xiao Hu, J Stephen Downie, Cyril Laurier, Mert Bay, and Andreas F Ehmann. The 2007 MIREX audio mood classification task: Lessons learned. In *Statistics*, pages 462–467. Citeseer, 2008.

Xiao Hu, J Stephen Downie, and Andreas F Ehmann. Lyric text mining in music mood classification. *Information Retrieval*, 183(Ismir):411–416, 2009a.

Yajie Hu, Xiaoou Chen, and Deshun Yang. Lyric-based song emotion detection with affective lexicon and fuzzy clustering method. *Information Retrieval*, (Ismir): 123–128, 2009b.

Arefin Huq, Juan Pablo Bello, and Robert Rowe. Automated Music Emotion Recognition: A Systematic Evaluation. *Journal of New Music Research*, 39(3):227–244, 2010. ISSN 09298215.

V Imbrasaitė and P Robinson. Absolute or relative? A new approach to building feature vectors for emotion tracking in music. In *Proc. ICME3*, 2013.

V Imbrasaitė, T Baltrusaitis, and P Robinson. Emotion tracking in music using Continuous Conditional Random Fields and relative feature representation. In *Proc. IEEE ICME*, 2013.

Dan-Ning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai. Music type classification by spectral contrast feature. *Proceedings. IEEE International Conference on Multimedia and Expo*, 2002.

P N Juslin. Cue utilization in communication of emotion in music performance: relating performance to perception. *Journal of Experimental Psychology: Human Perception and Performance*, 26(6):1797–1813, 2000.

P N Juslin and J A Sloboda. *Music and emotion: Theory and research*, volume 20 of *Series in Affective Science*. Oxford University Press, 2001. ISBN 9780192631886.

P N Juslin and J A Sloboda. *Music and Emotion: Theory, Research, Applications*. Oxford University Press, 2010.

K Kallinen and N Ravaja. Emotion perceived and emotion felt: Same and different. *Musicae Scientiae*, 10(2):191–213, 2006. ISSN 10298649.

Ittipan Kanluan, Michael Grimm, and Kristian Kroschel. Audio-visual emotion recognition using an emotion space concept. In *European Signal Processing Conference*, 2008.

Alistair Kennedy and Diana Inkpen. Sentiment classification of movie reviews using contextual valence shifters. In *Computational Intelligence*, volume 22, pages 110–125, 2006.

Youngmoo E Kim. Moodswings: a collaborative game for music mood label collection. *Computer Engineering*, pages 231–236, 2008.

Youngmoo E Kim, Donald S Williamson, and Sridhar Pilli. Towards quantifying the album effect in artist identification. In *Proc. of ISMIR*, pages 393–394, 2006.

S Koelsch, W A Siebel, and T Fritz. Chapter 12, Functional neuroimaging. In P N Juslin and J A Sloboda, editors, *Handbook of music and emotion theory research application*, pages 313–346. OUP, 2010.

M D Korhonen, D A Clausi, and M E Jernigan. Modeling emotional content of music using system identification. *IEEE transactions on systems man and cybernetics Part B Cybernetics*, 36(3), 2006. ISSN 10834419.

Carol L Krumhansl. An exploratory study of musical emotions and psychophysiology. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, 51(4):336, 1997.

Naveen Kumar, Rahul Gupta, Tanaya Guha, Colin Vaz, Maarten Van Segbroeck, Jangwon Kim, and Shrikanth S Narayanan. Affective Feature Design and Predicting Continuous Affective Dimensions from Music. In *MediaEval Workshop*, Barcelona, Spain, 2014.

J Lafferty, A McCallum, and F Pereira. Conditional random fields: probabilistic models for segmenting and labelling sequence data. In *ICML*, 2001.

Cyril Laurier, Jens Grivolla, and Perfecto Herrera. Multimodal Music Mood Classification Using Audio and Lyrics. *2008 Seventh International Conference on Machine Learning and Applications*, pages 688–693, 2008.

Tao Li and Mitsunori Ogihara. Detecting emotion in music. In Holger H Hoos and David Bainbridge, editors, *Proceedings of the International Symposium on Music Information Retrieval*, number 136091467180426693O3related:90Ab8dVm3bwJ, pages 239–240. Citeseer, 2003. ISBN 2974619401.

D Liu. Automatic mood detection and tracking of music audio signals. *IEEE Transactions on Audio, Speech and Language Processing*, 14:5–18, 2006. ISSN 1558-7916.

Dan Liu, Lie Lu, and Hong-Jiang Zhang. Automatic mood detection from acoustic music data. *Stress The International Journal on the Biology of Stress*, PAMI-8(6):81–7, 2003.

Antoine Liutkus, Zafar Rafii, Roland Badeau, Bryan Pardo, and Gaël Richard. Adaptive filtering for music/voice separation exploiting the repeating musical structure. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 53–56. IEEE, 2012.

B. Logan, A. Kositsky, and P. Moreno. Semantic analysis of song lyrics. *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, 2, 2004.

F Lotte, M Congedo, A Lécuyer, F Lamarche, and B Arnaldi. A review of classification algorithms for EEG-based computer interfaces. *Journal of Neural Engineering*, 4(2):R1–R13, 2007. ISSN 17412552.

K F MacDorman and Stuart Ough Chin-Chang Ho. Automatic Emotion Prediction of Song Excerpts: Index Construction, Algorithm Design, and Empirical Comparison. *Journal of New Music Research*, 36(4), 2007. ISSN 09298215.

Jose P G Mahedero, Álvaro MartÍnez, Pedro Cano, Markus Koppenberger, and Fabien Gouyon. Natural language processing of lyrics. *Structure*, 131(10):475–478, 2005.

Yi Mao and Guy Lebanon. Isotonic Conditional Random Fields and Local Sentiment Flow. *Advances in Neural Information Processing Systems 19*, 19(April 2008): 961–968, 2007. ISSN 10495258.

Konstantin Markov. Dynamic Music Emotion Recognition Using State-Space Models. In *MediaEval Workshop*, pages 5–6, Barcelona, Spain, 2014.

M. McVicar, Tim Freeman, and T. De Bie. Mining the correlation between lyrical correlation between lyrical and audio features and the emergence of mood. *12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, pages 783–788, 2011.

Thomas Minka and John Lafferty. Expectation-Propagation for the Generative Aspect Model. In *Uncertainty in Artificial Intelligence*, pages 352–359, 2002.

Tony Mullen and Nigel Collier. Sentiment analysis using support vector machines with diverse information sources. In Dekang Lin Wu and Dekai, editors, *Conference on Empirical Methods in Natural Language Processing*, pages 412–418. Association for Computational Linguistics, 2004.

Robert Neumayer and Andreas Rauber. Integration of Text and Audio Features for Genre Classification in Music Information Retrieval. *Advances in Information Retrieval: Lecture Notes in Computer Science*, (4425):724–727, 2007. ISSN 3540714944; 9783540714941.

Mihalis A Nicolaou, Hatice Gunes, and Maja Pantic. Continuous Prediction of Spontaneous Affect from Multiple Cues and Modalities in Valence-Arousal Space, 2011. ISSN 19493045.

Mihalis A Nicolaou, Hatice Gunes, and Maja Pantic. Output-associative rvm regression for dimensional and continuous emotion prediction. *Image and Vision Computing*, 30(3):186–196, 2012.

R Panda and R P Paiva. Using Support Vector Machines for Automatic Mood Tracking in Audio Music. In *130th Audio Engineering Society Convention*, 2011.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *Proc. 43st ACL*, 2005.

Bo Pang and Lillian Lee. Opinion Mining and Sentiment Analysis. *Foundations and Trends® in Information Retrieval*, 2(2):1–135, 2008. ISSN 15540669.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, number July, pages 79–86. cornell, 2002.

J Peng, L Bo, and J Xu. Conditional neural fields. *Advances in Neural Information Processing Systems*, 22, 2009.

Isabelle Peretz. Towards a neurobiology of musical emotions. In P N Juslin and J A Sloboda, editors, *Handbook of music and emotion: theory research application*, chapter 5, pages 99–126. Oxford University Press, 2010.

Rosalind W Picard. *Affective Computing*, volume 136. MIT Press, 1997. ISBN 0262161702.

John C Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. *Advances in Kernel MethodsSupport Vector Learning*, 208: 1–21, 1998.

Tao Qin, Tie-yan Liu, Xu-dong Zhang, De-sheng Wang, and Hang Li. Global Ranking Using Continuous Conditional Random Fields. In *NIPS*, 2008.

V. Radosavljevic, S. Vucetic, and Z. Obradovic. Continuous Conditional Random Fields for Regression in Remote Sensing. In *ECAI*, pages 809–814, 2010.

Zafar Rafii and Bryan Pardo. Music/voice separation using the similarity matrix. In *ISMIR*, number Ismir, pages 583–588, Porto, Portugal, 2012.

Zafar Rafii and Bryan Pardo. REpeating pattern extraction technique (REPET): A simple method for music/voice separation. *IEEE Transactions on Audio, Speech and Language Processing*, 21(1):71–82, 2013.

Geovany A Ramirez, Tadas Baltrušaitis, and Louis-Philippe Morency. Modeling latent discriminative dynamic of multi-dimensional affective signals. In *ACII*, Berlin, Heidelberg, 2011. Springer-Verlag.

James A Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, 1980. ISSN 00223514.

I Sato, K Kurihara, and H Nakagawa. Deterministic Single-Pass Algorithm for {LDA}. In *Advances in Neural Information Processing Systems 23 (NIPS)*, 2010.

Klaus R Scherer. The dynamic architecture of emotion: Evidence for the component process model. *Cognition & Emotion*, 23(7):1307–1351, 2009. ISSN 02699931.

KR Scherer. Emotion expression in speech and music. In Johan Sundberg, L Nord, and R Carlson, editors, *Music, language, speech and brain*, pages 146–156. London: Macmillan, 1991.

E M Schmidt and Y E Kim. Prediction of time-varying musical mood distributions from audio. In *Proc. of ISMIR*, pages 465–470, 2010a.

E M Schmidt and Y E Kim. Prediction of Time-Varying Musical Mood Distributions Using Kalman Filtering. *9th ICMLA*, pages 655–660, 2010b.

E M Schmidt and Y E Kim. Modeling musical emotion dynamics with Conditional Random Fields. *Proc. of ISMIR*, 2011a. ISSN 07307829.

E M Schmidt, D Turnbull, and Y E Kim. Feature selection for content-based, time-varying musical emotion regression. In *Proc. of ISMIR*. ACM, 2010. ISBN 9781605588155.

Erik M. Schmidt and Youngmoo E. Kim. Learning emotion-based acoustic features with deep belief networks. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 65–68, 2011b.

Erik M Schmidt, Matthew Prockup, Jeffery Scott, Brian Dolhansky, Brandon G Morton, and Youngmoo E Kim. Relating Perceptual and Feature Space Invariances in Music Emotion Recognition. In *9th International Symposium on Computer Music Modeling and Retrieval (CMMR)*, pages 19–22, London, UK, 2012.

E Schubert. Modeling Perceived Emotion With Continuous Musical Features. *Music Perception*, 21(4), 2004. ISSN 07307829.

Emery Schubert, Sam Ferguson, Natasha Farrar, David Taylor, and Gary E Mcpherson. Continuous Response to Music using Discrete Emotion Faces. In *Proceedings of Computer Music Modeling and Retrieval*, number June, pages 3–19, 2012.

Björn Schuller, Clemens Hage, Dagmar Schuller, and Gerhard Rigoll. Mister D.J., Cheer Me Up!': Musical and Textual Features for Automatic Mood Classification. *Journal of New Music Research*, 39(1):13–34, 2010. ISSN 09298215.

Björn Schuller, Felix Weninger, and Johannes Dorfner. Multi-modal non-prototypical music mood analysis in continuous space: reliability and performances. *Proceedings of International Symposium on Music Information Retrieval*, pages 759–764, 2011.

Björn Schuller, Michel Valstar, Roddy Cowie, and Maja Pantic. Avec 2012: the continuous audio/visual emotion challenge - an introduction. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, ICMI '12, pages 361–362. ACM, 2012. ISBN 978-1-4503-1467-1.

Björn Schuller, Stefan Steidl, Anton Batliner, Alessandro Vinciarelli, Klaus Scherer, Fabien Ringeval, Mohamed Chetouani, Felix Weninger, Florian Eyben, Erik Marchi, et al. The interspeech 2013 computational paralinguistics challenge: social signals, conflict, emotion, autism. 2013.

Jeffrey Scott, Erik M Schmidt, Matthew Prockup, Brandon Morton, and Youngmoo E Kim. Predicting Time-Varying Musical Emotion Distributions from Multi-Track Audio. In *9th International Symposium on Computer Music Modeling and Retrieval (CMMR)*, number June, pages 19–22, London, 2012.

Janto Skowronek, Martin F McKinney, and Steven Van De Par. A Demonstrator for Automatic Music Mood Estimation. In Simon Dixon, David Bainbridge, and Rainer Typke, editors, *Audio*, pages 345–346. Österreichische Computer Gesellschaft, 2007. ISBN 978385403218.

J A Sloboda and P N Juslin. At the interface between the inner and outer world: psychological perspectives. In P N Juslin and J A Sloboda, editors, *Handbook of music and emotions theory research applications*, chapter 4, pages 73–98. OUP, 2010.

Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression, 2004.

Mohammad Soleymani, Michael N Caro, Erik M Schmidt, and Yi-Hsuan Yang. The mediaeval 2013 brave new task: Emotion in music. In *MediaEval*. Citeseer, 2013a.

Mohammad Soleymani, Micheal N. Caro, Erik M. Schmidt, Cheng-Ya Sha, and Yi-Hsuan Yang. 1000 songs for emotional analysis of music. In *Proceedings of the 2Nd ACM International Workshop on Crowdsourcing for Multimedia*, CrowdMM '13, pages 1–6, New York, NY, USA, 2013b. ACM. ISBN 978-1-4503-2396-3.

Jacquelin A Speck, Erik M Schmidt, Brandon G Morton, and Youngmoo E Kim. A comparative study of collaborative vs. traditional music mood annotation. *Proc. of ISMIR*, 2011.

Stefan Steidl, Anton Batliner, Alessandro Vinciarelli, Klaus Scherer, Fabien Ringeval, Mohamed Chetouani, Felix Weninger, Florian Eyben, Erik Marchi, Marcello Mortillaro, Hugues Salamin, Anna Polychroniou, Fabio Valente, Samuel Kim, Machine Intelligence, and Signal Processing. The INTERSPEECH 2013 Computational Paralinguistics Challenge : Social Signals , Conflict , Emotion , Autism. In *Proc. of INTERSPEECH*, pages 148–152, Lyon, France, 2013.

C Sutton and A Mccallum. *Introduction to Conditional Random Fields for Relational Learning*. MIT Press, 2006.

Mike Thelwall, Kevan Buckley, Georgios Paltoglou, and Di Cai. Sentiment Strength Detection in Short Informal Text. *Journal of the American Society for Information Science*, 61(12):2544–2558, 2010. ISSN 15322882.

William Forde Thompson and Frank A Russo. The attribution of emotion and meaning to song lyrics. In *Polskie Forum Psychologiczne*, volume 9, pages 51–62, 2004.

Konstantinos Trohidis and George Kalliris. Multi-label classification of music into emotions. *Learning*, 2008:325–330, 2008.

George Tzanetakis. Marsyas submission to MIREX 2007. *Marsyas*, 53(2):151, 2007.

Dimitrios Ververidis and Constantine Kotropoulos. Emotional speech recognition: Resources, features, and methods. *Speech Communication*, 48(9):1162–1181, 2006.

Ju-Chiang Wang, Yi-Hsuan Yang, Hsin-Min Wang, and Shyh-Kang Jeng. The Acoustic Emotion Gaussians Model for Emotion-based Music Annotation and Retrieval Categories and Subject Descriptors. In *ACM International Conference on Multimedia (MM)*, pages 89–98, 2012.

M Wang, N Zhang, and H Zhu. User-adaptive music emotion recognition. *Proceedings 7th International Conference on Signal Processing 2004 Proceedings ICSP 04 2004*, pp(60174015):1352–1355, 2004.

Xing Wang, Xiaoou Chen, Deshun Yang, and Yuqian Wu. Music emotion classification of chinese songs based on lyrics using tf* idf and rhyme. In *ISMIR*, pages 765–770. Citeseer, 2011.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior research methods*, 45 (4):1191–207, 2013. ISSN 1554-3528.

Felix Weninger and Florian Eyben. The TUM Approach to the MediaEval Music Emotion Task Using Generic Affective Audio Features. In *MediaEval Workshop*, pages 10–11, Barcelona, Spain, 2013.

Theresa Wilson, Janyce Wiebe, and Rebecca Hwa. Just how mad are you? Finding strong and weak opinion clauses. *Science*, 04:761–769, 2004.

Martin Wöllmer, Florian Eyben, Stephan Reiter, Björn Schuller, Cate Cox, Ellen Douglas-Cowie, and Roddy Cowie. Abandoning emotion classes - towards continuous emotion recognition with modelling of long-range dependencies. In *INTERSPEECH*. ISCA, 2008.

Yunqing Xia, Linlin Wang, Kam-Fai Wong, and Mingxing Xu. Sentiment vector space model for lyric-based song sentiment classification. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies Short Papers HLT 08*, (June):133–136, 2008.

Dan Yang and Won Sook Lee. Disambiguating music emotion using software agents. In *Systems Science*, volume 4, pages 52–57. Universitat Pompeu Fabra, 2004. ISBN 8488042442.

Dan Yang and Won Sook Lee. Music emotion identification from lyrics. In *ISM 2009 - 11th IEEE International Symposium on Multimedia*, pages 624–629, 2009.

Yi-Hsuan Yang, Chia-Chu Liu, and Homer H Chen. Music emotion classification: a fuzzy approach. *Proceedings of the 14th annual ACM international conference on Multimedia*, 35(4):3–6, 2006.

Yi-Hsuan Yang, Ya-Fan Su, Yu-Ching Lin, and Homer H Chen. Music Emotion Recognition : The Role of Individuality. *Work*, pages 13–21, 2007.

Yi-Hsuan Yang, Yu-Ching Lin, Heng-Tze Cheng, I-Bin Liao, Yeh-Chin Ho, and Homer H Chen. Toward multi-modal music emotion classification. 5353:70–79, 2008.

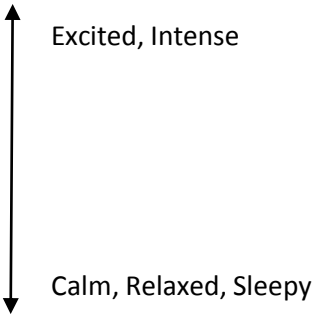
Zhihong Zeng, Maja Pantic, Glenn I Roisman, and Thomas S Huang. A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions. *TPAMI*, 31(1), 2009. ISSN 0162-8828.

M Zentner, D Grandjean, and K R Scherer. Emotions evoked by the sound of music: Differentiation, classification, and measurement. *Emotion*, 8(4):494–521, 2008.

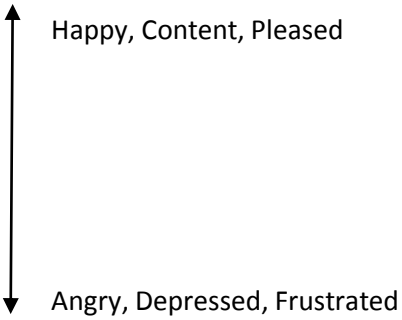
Appendix A

The following page shows the handout given to participants to help them understand and remember the two affective axes used in the experiment described in Chapter 3.

Study one helper:



Study two helper:



Study three helper: